ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

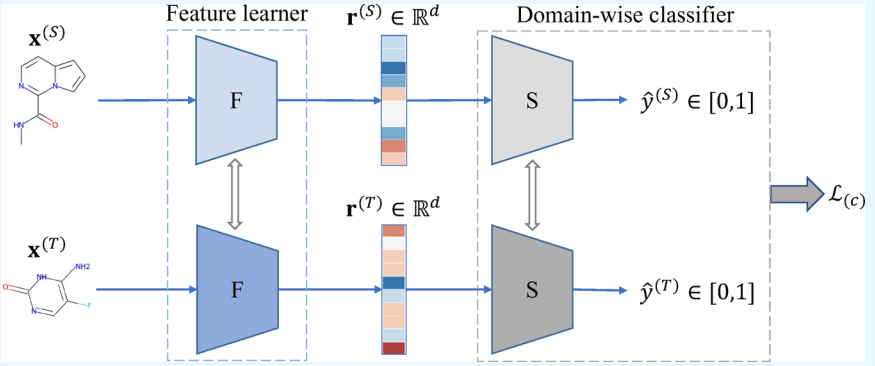
Improving Compound Activity Classification via Deep Transfer and Representation Learning

Vishal Dey, Raghu Machiraju, and Xia Ning*

Cite This: *ACS Omega* 2022, 7, 9465−9483

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

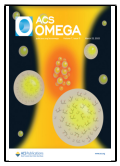


ABSTRACT: Recent advances in molecular machine learning, especially deep neural networks such as graph neural networks (GNNs), for predicting structure−activity relationships (SAR) have shown tremendous potential in computer-aided drug discovery. However, the applicability of such deep neural networks is limited by the requirement of large amounts of training data. In order to cope with limited training data for a target task, transfer learning for SAR modeling has been recently adopted to leverage information from data of related tasks. In this work, in contrast to the popular parameter-based transfer learning such as pretraining, we develop novel deep transfer learning methods *TAc* and *TAc-fc* to leverage source domain data and transfer useful information to the target domain. *TAc* learns to generate effective molecular features that can generalize well from one domain to another and increase the classification performance in the target domain. Additionally, *TAc-fc* extends *TAc* by incorporating novel components to selectively learn feature-wise and compound-wise transferability. We used the bioassay screening data from PubChem and identified 120 pairs of bioassays such that the active compounds in each pair are more similar to each other compared to their inactive compounds. Overall, *TAc* achieves the best performance with an average *ROC-AUC* of 0.801; it significantly improves the *ROC-AUC* of 83% of target tasks with an average task-wise performance improvement of 7.102%, compared to the best baseline *dmpna*. Our experiments clearly demonstrate that *TAc* achieves significant improvement over all baselines across a large number of target tasks. Furthermore, although *TAc-fc* achieves slightly worse *ROC-AUC* on average compared to *TAc* (0.798 vs 0.801), *TAc-fc* still achieves the best performance on more tasks in terms of *PR-AUC* and *F1* compared to other methods. In summary, *TAc-fc* is also found to be a strong model with competitive or even better performance than *TAc* on a notable number of target tasks.

1. INTRODUCTION

Drug discovery is a time-consuming and expensive process[1]—it takes at least 10 years and at least $1 billion to fully develop a drug.[2] During the initial stages of this process, promising drug candidates are identified by screening a large library of chemical compounds and then further investigated for specific properties. In order to speed up this process, computational approaches[3,4] have been adopted, particularly for identifying potential drug candidates during the initial stages of drug discovery. Computational approaches explore a much larger space of chemical compounds to predict their physio-chemical properties and/or biological activities toward the target. In this paper, we consider the problem of compound bioactivity classification, where a compound is classified as active or inactive based on whether that compound binds to the protein target. Biological activities of compounds are initially examined in a bioassay by measuring their binding affinities or dissociation constants toward the target. Significant research[5−7] has established the relationship between the chemical structures and biological activities of compounds, also known as structure−activity relationships (SARs).[5] Several computational approaches[8] have been developed to model SARs and to predict compound bioactivities

Received: December 1, 2021
Accepted: February 23, 2022
Published: March 11, 2022



https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

from their 2D/3D structures. However, most popular approaches such as deep neural networks require large amounts of labeled data for effective SAR modeling. Thus, the limited availability of bioassay data for specific targets still poses a major challenge in effective SAR modeling.[9]

Over the years, several methods[10−12] aimed to improve SAR predictions for specific targets by leveraging activity information from related targets. These methods consider targets to be related, based on the principles from chemogenomics.[13−16] *The key principle behind these methods is that similar proteins tend to bind to structurally similar compounds.* In this work, we consider proteins belonging to the same protein family to be similar. Thus, leveraging compound activity information from bioassays corresponding to a set of proteins from the same protein family (e.g., G-coupled protein receptors, kinases, peptidases, etc.) collectively might better inform the SAR model than the individual bioassays. In essence, transfer learning can enable better SAR modeling by leveraging information from such related bioassays. However, existing methods are instance-based transfer learning methods.[17] They select a subset of data from related bioassays and then augment the training data for the target task with the selected subset. Existing deep transfer learning-based methods[18] for SAR modeling are either parameter-based (such as fine tuning) or feature-based, out of which parameter-based methods are more popular. However, such methods can lead to overfitting and negative transfer,[17] especially when the targets are not related. In this regard, we believe that feature-based methods are better in that they can learn the similarity/relatedness between the targets in the latent space in a data-driven manner.

Primarily, we develop an instance-based transfer learning method *TAc* that leverages target information from related bioassays, based on the key principle of chemogenomics as mentioned earlier. We further extend *TAc* to novel feature-based deep transfer learning methods *TAc* and *TAc-fc* that quantitatively measure transferability and explicitly learn what to transfer in a fully data-driven manner. To this end, we develop novel components to learn feature-wise and compound-wise transferability in order to effectively encode the commonalities among compounds of different tasks. In order to represent compounds, we leveraged the popular idea of a directed message passing neural network (*dmpn*)[19] and added an attention-based pooling mechanism, denoted as *dmpna*. We collected a set of confirmatory bioassays from PubChem[20] that have a single protein target and are tested on chemical substances. We identified 120 bioassay pairs involving 59 protein targets such that the active compounds in each pair are more similar to each other compared to the inactive compounds. We compared our methods *TAc* and *TAc-fc* with several baselines with respect to two aspects: compound representation and transfer mechanisms. Overall, *TAc-dmpna* achieves the best performance compared to all other methods. Compared to *TAc-dmpna*, *TAc-fc-dmpna* performs slightly worse, but the latter still provides significant performance improvement on some target tasks. This suggests that although the transfer mechanism in *TAc* performs the best overall, the deep transfer mechanism with learned feature-wise and compound-wise transferability can actually benefit some targets. Furthermore, experimental results demonstrate the efficacy of our proposed attention mechanism of *dmpna* in learning better compound features. We provide additional experiments on the compound prioritization problem[12] where *dmpna* clearly outperforms all other compound representation methods.

The rest of the paper is organized as follows. Section 1.1 presents the related works in drug discovery and transfer learning with applications in SAR predictions. Section 2 presents the materials used for experimental evaluation, experimental results, and detailed analyses with discussions. Section 3 presents the conclusions, and Section 4 presents the notations and definitions used in this paper and the proposed methods of transfer learning for activity prediction.

1.1. Related Work. In this section, we provide a brief overview of existing works and divide them across three subsections as follows. In Section 1.1.1, we summarize notable works on computational approaches in drug discovery. In Section 1.1.2, we provide a brief overview of existing works in transfer learning. In Section 1.1.3, we provide an overview of existing methods that use transfer learning for better SAR modeling.

1.1.1. Computational Methods in Drug Discovery. The first step in the drug discovery process is to conduct bioassays[21] that screen a large set of compounds for desirable properties (e.g., activity, solubility, and toxicity). The findings from these bioassays guide the later steps of the drug discovery process. In order to speed up initial stages of the drug discovery process, computational approaches have been adopted. Computational approaches to predict activities/properties of compounds from their molecular structures have been a significant research area in cheminformatics.[8,22,23] These approaches rely on the quantitative structure−activity/property relationship (QSAR/QSPR)[5,24] to predict compound activities/properties as expressed in bioassays.

In order to predict such activities/properties, machine learning methods such as classification and regression are typically used. Binary/real-valued observations from bioassay data are used to train these classification/regression methods. Popular conventional classification and regression methods to predict compound activities/properties consist of support vector machines,[25−28] random forests,[29,30] Bayesian models,[31,32] etc. In these methods, compounds are typically represented by hand-crafted molecular fingerprints[33,34] or descriptors.[35] Recently, deep learning methods[36−40] have demonstrated significant performance improvement over conventional methods across several activity/property prediction tasks.[41−43] Unlike conventional methods, these methods do not require careful and expensive design of hand-crafted molecular fingerprints or descriptors by domain experts. These methods learn the compound representations from molecular graphs[19,44−48] and SMILES strings,[49−51] in a fully data-driven manner for each task. Such learned representations are task-specific and can better encode relevant structures for each task. Thus, such learned representations are often more effective than molecular fingerprints or descriptors. While these deep learning models have achieved the state-of-the-art performance on several molecular activity/property prediction tasks, these models require a large amount of labeled training data to encode relevant patterns into learned representations. Training these models with limited labeled data for certain prediction tasks often leads to subpar performance.

1.1.2. Transfer Learning. In order to effectively train models with limited labeled data for certain prediction tasks, transfer learning between related tasks has been widely explored in Computer Vision (CV) and Natural Language Processing (NLP).[52,53] Transfer learning[17] is an emerging research area in which knowledge gained from auxiliary tasks is transferred to improve the predictive performance of the target task. Instead of

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

training a model for the target task from scratch, a popular transfer learning technique, called fine-tuning,[54] fine-tunes the model pretrained from other related tasks. Pretraining does not explicitly learn what/when to transfer and rather relies on the model parameters to encode and transfer information across different tasks. Although pretraining is the most popular transfer learning method, it does not guarantee improvement (due to "negative transfer"[55]). Moreover, fine-tuning a highly parametrized model with limited data may lead to overfitting to the training data, and thus, the fine-tuned model might not generalize well to the test data. Apart from pretraining, another area of deep transfer learning, called domain adaptation, has gained a lot of attention.[56−58] Domain adaptation methods reduce the effect of the domain shift by learning domain-invariant representations that can generalize well across different tasks. In order to learn such representations, domain adaptation methods either minimize statistical measures[59−61] of domain shift or use adversarial training.[62] Following the success of adversarial training in generative adversarial networks (GANs),[63] adversarial domain adaptation methods[64−66] gained more attention and demonstrated state-of-the-art performances over benchmark CV and NLP data sets. Adversarial domain adaptation methods use adversarial training to learn domain-invariant representations via a minimax optimization using a feature extractor, a domain classifier, and a label predictor. The principle of adversarial training is used to train the feature extractor to learn domain-invariant representations which are indistinguishable by the domain classifier. Seminal methods in adversarial domain adaptation[64,65,67] differ in the design choices, such as adversarial loss functions, optimization, coupling of weights, etc. Other existing methods focus on conditional feature alignment,[68,69] multisource transfer,[70,71] etc. However, these methods have been specifically developed for image domain adaptation or image translation problems. To the best of our knowledge, none of these methods have been widely adapted for graph-structured data. In this work, following the idea of adversarial domain adaptation, we proposed a novel transfer learning method that learns effective compound representations from graph-structured data and transfers relevant information from a related task to the target task.

1.1.3. Transfer Learning in SAR Predictions. To alleviate the limited data problem in cheminformatics, various transfer learning[18,72−75] and multitask learning methods[76−80] have been recently developed. Inspired by the success of pretraining followed by fine-tuning in CV and NLP, Goh et al.[81] proposed ChemNet, where a deep neural network is pretrained on a large set of compounds in a self-supervised manner and then fine-tuned on individual activity/prediction tasks. Following the same idea, Li and Fourches[82] proposed MolPMoFit which trains a long short-term memory (LSTM)[83] on SMILES strings of compounds and then fine-tunes the pretrained model on specific tasks. Although pretraining has been widely studied, existing work in cheminformatics does not demonstrate significant performance improvement over the state-of-the-art supervised models in a single-task setting. Moreover, models trained on SMILES strings do not explicitly leverage the topological information of compounds. However, our methods use molecular graphs as inputs and hence explicitly leverage the topological information.

Adversarial transfer learning has been rarely explored for SAR predictions and on graph-structured data. To the best of our knowledge, only recently Abbasi et al.[84] combined multitask networks and adversarial domain adaptation to learn transferable molecular representations from multiple-source bioassays to improve the prediction performance on the target bioassay. The authors evaluated their model on biophysics and physiology data sets such as Tox21, SIDER, BACE, ToxCast, and HIV. Experimental results demonstrated that the proposed method outperforms no-transfer methods only on a few target tasks. Moreover, experimental results do not clearly demonstrate the contribution of the adversarial domain adaptation component to the overall performance. Overall, prior work on transfer-learning-based SAR modeling does not clearly suggest a performance gain over conventional SAR models over a wide array of target tasks.

2. RESULTS AND DISCUSSION

In this section, we present the materials used for experimental evaluation (Section 2.1), followed by detailed experimental results and discussions (Sections 2.2−2.7).

2.1. Materials. In this section, we describe the data set generation, baseline methods, and experimental protocols in detail.

2.1.1. Data Set Generation. We used the real screening data from PubChem to test our methods. PubChem[20,85] is one of the largest public chemical databases with more than 271 M substances, 111 M unique chemical structures, and 293 M bioassay data. We selected a set of bioassays from PubChem bioassays [accessed on 2020-12-25] such that each bioassay has a sufficiently large number of active and inactive compounds. Then, we generated pairs of bioassays for transfer learning in accordance to the protocols in below Sections 2.1.1.1 and 2.1.1.2.

2.1.1.1. Initial Bioassay Selection and Pruning. We first selected a set of 7284 confirmatory bioassays that have a single protein target and are tested on chemical substances. These bioassays have 1279 unique protein targets in total. Among these protein targets, we were able to identify the organism and protein family information for 961 protein targets within 435 protein families using UniProt.[86] Among the 435 protein families, we further combined them into 278 families (e.g., Peptidase A1, Peptidase C12, and Peptidase C13 families were combined into the peptidase family). Among the 278 families, we selected 10 that have the most protein targets belonging to "Human" organisms. These top 10 protein families are the (1) G-protein-coupled receptor 1 family, (2) peptidase family, (3) protein kinase family, (4) nuclear hormone receptor family, (5) protein-tyrosine phosphatase family, (6) ABC transporter family, (7) cytochrome P450 family, (8) Bcl-2 family, (9) G-protein-coupled receptor 3 family, and (10) histone deacetylase family. These protein families involved 269 unique protein targets and covered the major drug targets in drug discovery.[87,88]

According to the 10 protein families, bioassays with targets from these protein families were then processed as follows:

1. We combined bioassays of the same target into one bioassay, resulting in 269 combined bioassays.
2. For each combined bioassay, we selected its compounds that were tested for inhibition against the target (i.e., the corresponding PubChem activity type specified by the depositor was "inhibitor").
3. From those inhibitive compounds, we selected the compounds that were specified as either "active" or "inactive" against the target and discarded the compounds that were specified as "inconclusive" or "undetermined".

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

4. If the active/inactive compounds appeared multiple times in the bioassay with the same activity label, we retained one of their records. If the active/inactive compounds appeared multiple times in the bioassay with different activity labels, we removed the compounds from the bioassays (in our data set, only about 2.08% of compounds for each bioassay on average appear multiple times with different activity levels). We use canonical SMILES strings to detect identical compounds.

After the above processing, each combined bioassay has on average 17 005 unique compounds in total, with 188 active and 16 817 inactive. Furthermore, out of the 269 combined bioassays, 95 bioassays have more than 50 active compounds. Among the 10 protein families involved in the 95 bioassays, 2 protein families had only 1 target with more than 50 active compounds. Thus, we removed these 2 protein families and only used the remaining 8 protein families and their 93 bioassays. This set of 93 bioassays has on average 40 115 compounds, with 521 active and 39 595 inactive. This set of bioassays will be used to create bioassay pairs as will be described in the next section. Table S1 in the Supporting Information presents the statistics of each of the 93 bioassays.

2.1.1.2. Transferable Bioassay Pairing. From the 93 processed bioassays, we constructed 765 bioassay pairs such that in each pair the protein targets of the two bioassays are from the same protein family. We selected targets from the same protein family because based on the key intuition of chemical genomics[15,16]—proteins from the same family tend to have similar binding pockets and bind to similar compounds—this is the physicochemical foundation to enable possible information transfer across protein targets, and such targets and their bioassays can be used to test transfer learning. We first ensured that each of the 765 pairs of bioassays had balanced active and inactive compounds as follows:

(1) In each pair of bioassays, we removed the compounds that appeared in both bioassays but with different activity labels (on average, 2.09% of all unique compounds in a pair of bioassays). This is to avoid any conflicting information across bioassays, which could adversely affect our transfer learning method.
(2) For compounds with the same activity labels in both the bioassays (on average, 1.82% of all unique compounds in a pair of bioassays), we randomly sampled half of them into one of the bioassays and the other half into the other. This is to avoid duplication of compounds across bioassays, which could lead to overestimation of predictive performance.
(3) After the above steps, for each bioassay of a pair, we used all its active compounds and randomly sampled the same number of inactive compounds. If the inactive compounds were not sufficient, we randomly sampled compounds from PubChem that were not active in the bioassay as additional inactive compounds for the bioassays. This is to ensure that each bioassay in a pair has an equal number of active and inactive compounds, and thus the learning will not be dominated by either active or inactive compounds. Please note that a bioassay involved in two pairs may have different numbers of active and inactive compounds due to its pairing to the other bioassay.

After the above steps, we selected the bioassay pairs such that each bioassay in each pair had at least 50 active compounds retained. There were 635 such pairs that involved 92 bioassays in total. Among the 635 pairs of bioassays, we further selected the pairs as follows, such that the active compounds in each pair are similar to each other compared to their inactive compounds:

1. For a pair of bioassays B_i and B_j and their respective active and inactive compounds, denoted as $\mathcal{X}^+_{B_i}$, $\mathcal{X}^-_{B_i}$, $\mathcal{X}^+_{B_j}$ and $\mathcal{X}^-_{B_j}$, respectively, we calculated the following two types of average compound similarities using the Taminoto coefficient[89] over Morgan-count fingerprints (with radius = 3 and dimension = 2048): (1) among compounds of the same labels across the two bioassays $\mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j})$ and $\mathrm{sim}(\mathcal{X}^-_{B_i}, \mathcal{X}^-_{B_j})$ and (2) among compounds of different labels across the two bioassays $\mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^-_{B_j})$ and $\mathrm{sim}(\mathcal{X}^-_{B_i}, \mathcal{X}^+_{B_j})$.

2. Based on the similarities, we selected a set of bioassay pairs, denoted as $\mathcal{P}_0$, such that in each pair the active compounds of the two bioassays are more similar; that is

$$\mathcal{P}_0 = \{(B_i, B_j) | \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) > \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^-_{B_j}) \text{ and } \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) > \mathrm{sim}(\mathcal{X}^-_{B_i}, \mathcal{X}^+_{B_j})\}$$

We identified 329 such pairs. From $\mathcal{P}_0$, we further selected a set of bioassay pairs, denoted as $\mathcal{P}$, such that in each pair the active compounds in the two bioassays have a similarity above a certain threshold, that is,

$$\mathcal{P} = \{(B_i, B_j) | (B_i, B_j) \in \mathcal{P}_0,\ \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) - \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^-_{B_j}) + \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) - \mathrm{sim}(\mathcal{X}^-_{B_i}, \mathcal{X}^+_{B_j}) \geq 0.026\}$$

where 0.026 is the average value among all these pairs $\mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) - \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^-_{B_j}) + \mathrm{sim}(\mathcal{X}^+_{B_i}, \mathcal{X}^+_{B_j}) - \mathrm{sim}(\mathcal{X}^-_{B_i}, \mathcal{X}^+_{B_j})$

After the above process, we identified 120 pairs of bioassays in $\mathcal{P}$, involving 59 bioassays and 7 protein families with 278 active and 278 inactive compounds in each bioassay on average. Table S2 presents all the pairs and their compound statistics.

2.1.2. Baseline Methods. We tested our *TAc* and *TAc-fc* methods with respect to two aspects: (1) compound representations and (2) transfer mechanisms. Compound representation is key to revealing information among compounds that can be leveraged to transfer across. Transfer machenisms are critical to enable effective transfer of revealed information across bioassays.

2.1.2.1. Compound Representation Methods. Specifically, we compared our compound representation method *dmpna* (i.e., the feature learner in Section 4.2.2) with the following compound representation methods:

- Binary Morgan fingerprint (*morgan*):[33] *morgan* uses a binary feature vector to present a compound, in which each dimension of the feature vector corresponds to a predefined substructure, and the binary value in that dimension represents if the compound has that substructure or not.
- Morgan count fingerprints (*morgan-c*):[33] *morgan-c* is very similar to *morgan* except that the values in *morgan-c* represent how many corresponding substructures the compound has.

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

- Directed Message Passing Network (*dmpn*):[19] The *dmpn* method learns molecular structures by passing messages along directed edges over molecular graphs. It produces two representations for each bond through message passing through the two directions along the bond. Then it learns atom representations from the incoming bond representations and generates a compound representation using mean pooling over the atom representations. The *dmpn* (https://github.com/chemprop/chemprop) method is the state-of-the-art compound embedding a learning approach for compound property prediction.

We generated *morgan* and *morgan-c* (with radius = 3 and size = 2048) using RDKit.[90] In order to only compare the different compound representation methods, not the transfer learning mechanisms, we used a two-layer fully connected network as the classifier S over the above baseline feature representations to predict activity labels. We used cross-entropy as the loss function for these baseline methods. The corresponding methods are denoted as *FCN-morgan*, *FCN-morganc*, and *FCN-dmpn*, respectively. Note that these three baseline methods do not have information transfer mechanisms—they are single-task compound prediction methods.

2.1.2.2. Learning Methods for Compound Prediction. We compared *TAc* and *TAc-fc* with a transfer learning baseline known as domain-adversarial neural network, denoted as *DANN*.[64] We selected *DANN* because, to the best of our knowledge, there are no existing transfer learning methods over graph-structured data, and *DANN* is a standard transfer learning baseline method used on other data (e.g., images).[58] In particular, we adapted *DANN* to learn compound features from graph-structured data via *GNN* (e.g., *dmpn* or *dmpna*).

DANN consists of three components: (1) a feature extractor that represents compounds via feature learning; (2) a label predictor that predicts activity labels from learned compound features; and (3) a domain classifier that discriminates between the source and target compounds during training. *DANN* learns compound features that can generalize well from one domain to another, such that the learned features contain little discriminative domain information and enable *DANN* to accurately predict activity labels.

The objective function in *DANN* consists of two losses: domain classification loss and label prediction loss. *DANN* uses a minimax optimization such that the domain classification loss is minimized with respect to the domain classifier and is maximized with respect to the feature extractor. Specifically, minimizing the domain classification loss will encourage the domain classifier to correctly discriminate between the source and target compounds. On the other hand, maximizing the domain classification loss will encourage the learning of generalizable compound features.

The feature learner and discriminators in *TAc-fc* are learned via a minimax optimization, similar to how the feature extractor and the domain classifier in *DANN* are learned. However, *TAc-fc* is different from *DANN* in that *TAc-fc* learns feature-wise transferability and compound-wise transferability, while *DANN* only learns compound-wise transferability. Furthermore, following Ganin et al.,[64] *DANN* is trained on labeled data from the source domain and unlabeled data from the target domain.

2.1.3. Experimental Protocols. 2.1.3.1. Experimental Settings. In our experiments, we split each of the target bioassay in a pair into 10 folds. For the target bioassay, we used 1 fold for modeling training, 1 fold for validation, and remaining 8 folds for testing. We performed the above process 10 times, with a different training fold each time, and reported the average performance over the test folds. The above 1:1:8 training/validation/testing ratio follows a typical setting in transfer learning,[91] where it is assumed that the training data are limited, so it is needed to leverage other tasks via transfer. We used this cross-validation setting because we did not have a benchmark test set for each bioassay, and a 10-fold cross-validation will reduce the variance of the model performance. When we transferred the information from the source bioassay to the target bioassay, we used all the folds of the source bioassay and the training fold of the target bioassay in *TAc* in order to maximize the information content in the source bioassay that could be leveraged.

If the baseline methods do not have an information transfer mechanism (e.g., *FCN-morgan*), we applied an additional setting to simulate information transfer: in addition to the target task's $\mathcal{T}^{(T)}$ own training compounds, we also used all the compounds from the source task $\mathcal{T}^{(S)}$ as training data of $\mathcal{T}^{(T)}$. Thus, the $\mathcal{T}^{(S)}$ compounds will enrich the $\mathcal{T}^{(T)}$ training data and bring (i.e., transfer) information from *Src* directly to *Tgt*. This setting is referred to as data transfer, denoted as DT. If we only use the $\mathcal{T}^{(T)}$ compounds for training as in conventional single-task models, this setting is denoted as *noT*.

We trained each model using an ADAM[92] optimizer with an initial learning rate of 10^{-3}. All the models are trained up to 40 epochs. We used a grid search to tune all the hyper parameters such as the dimension d of the compound embedding $\mathbf{r}$, hidden-layer dimension of the attention layer for *dmpna*, hidden-layer dimension in L and G, and batch size. We used the validation set to determine the optimal number of epochs. During training, we evaluated the performance of each model on the validation set at every epoch and chose the trained model at some epoch k that gives the best performance on the validation set; thus, we selected k as the optimal number of epochs. We used the *ROC-AUC* metric for the above performance evaluation. All evaluation metrics are discussed in the following section. All the hyper parameters are reported in Table S3 for reproducibility purposes.

2.1.3.2. Evaluation Metrics. We used the following evaluation metrics: area under the precision−recall curve (*PR-AUC*), area under the receiver operating characteristic curve (*ROC-AUC*), *precision*, *sens*, *accuracy*, and *F1* score.

- Area under the precision−recall curve (*PR-AUC*): A precision−recall curve is generated by (precision, recall) value pairs corresponding to variable thresholds. *PR-AUC* measures the area under the precision−recall curve and provides an aggregate measure of performance across all possible thresholds.
- Area under the receiver operating characteristics curve (*ROC-AUC*): A receiver operating characteristic (ROC) curve is generated by true positive rates against false positive rates at various threshold values. *ROC-AUC* measures the area under the ROC curve.
- *precision*: it is the ratio of correctly predicted positive instances out of all predicted positive instances (e.g., the ratio of predicted active compounds that are truly active).
- *sens*: it is the ratio of correctly predicted positive instances out of all ground-truth positive instances (e.g., the ratio of active compounds that are correctly predicted as active).

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

Table 1. Overall Comparison[a]

method	*ROC-AUC*	*PR-AUC*	*precision*	*sens*	*accuracy*	*F1*
FCN-morgan	0.727 ± 0.124	0.729 ± 0.121	0.648 ± 0.104	0.742 ± 0.131	0.661 ± 0.110	0.683 ± 0.105
FCN-morganc	0.731 ± 0.120	0.730 ± 0.118	0.653 ± 0.102	0.735 ± 0.132	0.664 ± 0.107	0.682 ± 0.105
FCN-dmpn	0.754 ± 0.101	0.733 ± 0.102	0.619 ± 0.116	0.739 ± 0.156	0.656 ± 0.087	0.655 ± 0.126
FCN-dmpna	0.755 ± 0.112	0.729 ± 0.112	0.660 ± 0.119	0.712 ± 0.165	0.665 ± 0.101	0.651 ± 0.136
FCN-dmpn (*DT*)	0.754 ± 0.104	0.735 ± 0.105	0.687 ± 0.106	0.686 ± 0.213	0.669 ± 0.088	0.655 ± 0.140
FCN-dmpna (*DT*)	0.763 ± 0.108	0.745 ± 0.109	0.702 ± 0.108	0.671 ± 0.213	0.672 ± 0.092	0.645 ± 0.148
DANN-dmpn	0.733 ± 0.103	0.715 ± 0.103	0.671 ± 0.110	0.647 ± 0.215	0.649 ± 0.084	0.623 ± 0.144
DANN-dmpna	0.734 ± 0.102	0.716 ± 0.104	0.676 ± 0.106	0.653 ± 0.226	0.651 ± 0.085	0.624 ± 0.154
TAc-dmpn	0.798 ± 0.103	0.785 ± 0.108	0.729 ± 0.095	0.729 ± 0.146	**0.721 ± 0.093**	0.714 ± 0.108
TAc-fc-dmpn	0.798 ± 0.102	0.784 ± 0.107	0.729 ± 0.094	**0.731 ± 0.142**	0.720 ± 0.091	**0.715 ± 0.102**
TAc-dmpna	**0.801 ± 0.102**	**0.786 ± 0.107**	**0.731 ± 0.094**	0.729 ± 0.143	0.720 ± 0.090	0.713 ± 0.103
TAc-fc-dmpna	0.798 ± 0.105	0.785 ± 0.109	0.730 ± 0.097	0.728 ± 0.147	0.719 ± 0.095	0.713 ± 0.109

[a]In this table, the columns *ROC-AUC*, *PR-AUC*, *precision*, *sens*, *accuracy*, and *F1*-score have the average and standard deviation over all bioassays in each performance metric. The best performance values are bold. The second best performance values are underlined.

Table 2. Performance Comparison of *TAc-dmpna* vs *FCN-dmpna*[a]

method	*ROC-AUC*	*PR-AUC*	*precision*	*sens*	*accuracy*	*F1*
TAc-dmpna	0.801	0.786	0.731	0.729	0.720	0.713
FCN-dmpna (*DT*)	0.763	0.745	0.702	0.671	0.672	0.645
diff %	4.980	5.503	4.131	8.644	7.143	10.543
t-diff %	5.702	6.085	4.876	25.281	7.727	18.464
	(2.80×10^{-19})	(8.00×10^{-21})	(1.69×10^{-11})	(1.73×10^{-09})	(1.19×10^{-29})	(8.93×10^{-20})
N-impv	199 (83%)	192 (80%)	157 (65%)	153 (64%)	201 (84%)	198 (82%)
t-impv %	7.102	8.044	9.293	44.261	9.509	23.532
	(5.56×10^{-22})	(5.51×10^{-26})	(2.81×10^{-27})	(7.36×10^{-25})	(5.02×10^{-35})	(3.60×10^{-26})

[a]In this table, the first two rows have the performance from respective methods averaged over all bioassays in each performance metric. The row *diff* % has the percentage difference of average performance in each metric from *TAc-dmpna* over *FCN-dmpna* (*DT*). The row *t-diff* % has the average of task-wise percentage improvement from *TAc-dmpna* over *FCN-dmpna* (*DT*) in respective metrics, with the corresponding *p*-value in parentheses below. The row *N-impv* has the number and percentage of target tasks where *TAc-dmpna* performs better than *FCN-dmpna* (*DT*) in respective metrics. The row *t-impv* % has the average of task-wise percentage improvement only among the corresponding improved tasks, with corresponding *p*-values in parentheses below.

- *accuracy*: it is the ratio of correctly predicted (positive and negative) instances out of all instances (e.g., the ratio of compounds that are correctly predicted as active/inactive).
- *F1*-score: it is the harmonic mean of *precision* and *sens*.

If the above metrics have higher values, they indicate better performance.

2.1.4. Data and Software Availability. All the data sets and source code are publicly available at https://github.com/ninglab/TransferAct.

2.2. Overall Performance. Table 1 presents an overall performance comparison between *TAc-dmpn*, *TAc-fc-dmpn*, *TAc-dmpna*, *TAc-fc-dmpna*, and the baselines. The columns have the average and standard deviation over all bioassays in respective evaluation metrics achieved by the optimal models. Note that for each bioassay the optimal model of each method is the model that gives the best *ROC-AUC* value, and thus the performance of each method in other metrics does not necessarily correspond to the optimal performance in those metrics.

Table 1 shows that, overall, *TAc-dmpna* achieves the best performance compared to all other methods. Specifically, *TAc-dmpna* achieves the best average *ROC-AUC*, *PR-AUC*, and *precision* scores of 0.801, 0.786, and 0.731, respectively. This demonstrates that *TAc-dmpna* can learn effective compound features for the target task by leveraging source bioassay data and correctly predicts the compounds of the target bioassay. Furthermore, all variants of *TAc* and *TAc-fc*, especially *TAc-dmpn*, *TAc-fc-dmpn*, and *TAc-fc-dmpna*, achieve similar performance on average across all metrics. The performance of these three methods is not significantly different in most metrics. This suggests that learning feature-wise and compound-wise transferability via *TAc-fc* methods does not necessarily provide a performance boost on average. However, compared to the best method, *TAc-dmpna*, *TAc-fc-dmpn*, and *TA-fc-dmpna* improve *ROC-AUC* scores of 62% and 39% target tasks, respectively. On the whole, all variants of *TAc* and *TAc-fc* significantly outperform all baselines. Specifically, *TAc-dmpna* improves the average *ROC-AUC* by 4.9−10.1% and significantly improves *ROC-AUC* of at least 83% of the target tasks compared to any baseline method. Each of the other variants such as *TAc-dmpn*, *TAc-fc-dmpn*, and *TAc-fc-dmpna* improves *ROC-AUC* of at least 79% of the target tasks compared to any baseline method. This indicates that these methods can effectively transfer relevant information from the source task to the target task. In particular, the transfer learning mechanism in all variants of *TAc* and *TAc-fc* can better leverage source domain data compared to the transfer learning mechanism in other baselines. This is because both *TAc* and *TAc-fc* variants can better control the transferable information by incorporating varying degrees of task relatedness between the source and target tasks during training. Additionally, *TAc-fc*

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

variants can better extract relevant information from source domain data by learning feature-wise (Section 4.2.5.1) and compound-wise transferability (Section 4.2.5.2).

The best performance among the baseline methods is achieved by *FCN-dmpna* (*DT*). Table 2 presents the performance comparison between *TAc-dmpna* and *FCN-dmpna* (*DT*). The *diff* % values in Table 2 are calculated as the percentage difference of average performance in each metric from *TAc-dmpna* over *FCN-dmpna* (*DT*), where the average performance in each metric is calculated as the performance in that metric averaged over all the bioassays. The *t-diff* % values are calculated as the average of task-wise performance improvement (in %) from *TAc-dmpna* over *FCN-dmpna* (*DT*). The *N-impv* values denote the number of improved target tasks where *TAc-dmpna* performs better than *FCN-dmpna* (*DT*) in respective metrics. Considering only these *N-impv* improved tasks, the average of task-wise performance improvement (in %) is listed as *t-impv* % values. Similar to *t-diff* %, the numbers presented in parentheses in this row are the corresponding *p*-values for *t-impv* %. A *p-value* less than 0.05 was considered to be statistically significant.

Clearly, compared to the best baseline method *FCN-dmpna* (*DT*), *TAc-dmpna* improves the average *ROC-AUC*, *PR-AUC*, *precision*, *sens*, *accuracy*, and *F1* scores by 4.980%, 5.503%, 4.131%, 8.644%, 7.143%, and 10.543%, respectively. Furthermore, the average task-wise performance difference (i.e., *t-diff*%) from over *FCN-dmpna* (*DT*) across each metric is 5.702%, 6.085%, 4.876%, 25.281%, 7.727%, and 18.464%, respectively, and these differences are positive and statistically significant (as indicated by their corresponding *p*-values in parentheses), hence suggesting that the task-wise performance is significantly improved over *FCN-dmpna* (*DT*) . In particular, *TAc-dmpna* significantly improves the *ROC-AUC* performance of 199 out of 240 (83%) target tasks with an average task-wise improvement (i.e., *t-impv* %) of 7.102% (*p*-value: 5.56×10^{-22}). Such consistent and significant improvement (demonstrated by *t-impv* % and their corresponding *p*-values) across all evaluation metrics on a large percentage of target tasks (demonstrated by *N-impv*) provides strong evidence that *TAc-dmpna* clearly outperforms *FCN-dmpna* (*DT*) on the majority of target tasks. This further implies that the transfer mechanism in *TAc-dmpna* is more effective than that in *FCN-dmpna* (*DT*) . While *FCN-dmpna* (*DT*) pays equal attention to both the source and target tasks during training, *TAc-dmpna* can differentially focus on the two tasks by varying the weightage on the source classification loss (i.e., the trade-off parameter α in eq 3). Note that *TAc-dmpna* with $\alpha = 1$ is methodologically equivalent to *FCN-dmpna* (*DT*). By varying α, *FCN-dmpna* (*DT*) can incorporate different degrees of task relatedness between the source and target tasks during training. If the two tasks are not that related, a lower α will encourage the learning to focus more on the target task. In essence, learned compound features are more specific to the target task. On the other hand, α as high as 1 will enforce learning of compound features that generalize well across the two tasks. Such features may encode little target task-specific information and, hence, are not effective.

Furthermore, our experimental results in Table 1 demonstrate the efficacy of our proposed attention mechanism of *dmpna* in learning better compound features. Overall, both *dmpna*-based methods (i.e., *FCN-dmpna* and *FCN-dmpna* (*DT*)) outperform *dmpn*-based methods (i.e., *FCN-dmpn* and *FCN-dmpn* (*DT*)). Particularly, compared to *FCN-dmpn* (*DT*), *FCN-dmpna* (*DT*) improves about half of the target tasks of *ROC-AUC* of 152 out of 240 (63%) target tasks and gives significant performance improvement of 3.443% (*p*-value: 2.64×10^{-18}) on those improved target tasks. This demonstrates that the proposed attention mechanism in *dmpna* enables more effective compound features since it can differentially score atoms based on their relevance toward the final task. However, *FCN-dmpna* achieves either similar or slightly worse performance compared to *FCN-dmpn*. This is because *dmpna* with slightly more parameters than *dmpn* may struggle to capture relevant patterns during training, and thus *FCN-dmpna* can easily overfit to limited training data of the target task. In essence, this can lead to poor generalization performance on the test data. On the other hand, *FCN-dmpna* (*DT*) can generalize well since it is trained on the labeled source data along with the limited target data. Overall, the attention mechanism can better learn and effectively score the atoms in *FCN-dmpna* (*DT*) but not in *FCN-dmpna*, thereby achieving significant improvement in the former over *FCN-dmpn* (*DT*) and marginal improvement in the latter over *FCN-dmpn*. We will further demonstrate the efficacy of our proposed *dmpna* in the compound prioritization problem detailed in Section 2.7.

Furthermore, all *GNN*-based baselines (i.e., *FCN-dmpn*, *FCN-dmpna*, *FCN-dmpn* (*DT*), and *FCN-dmpna* (*DT*)) significantly outperform *DANN*-based methods. Our experimental results show that both *DANN*-based methods yield poor or similar performance compared to all other baseline methods. Specifically, the best *DANN* method (i.e., *DANN-dmpna*) reduces the average performance by 3−4% over the best baseline method *FCN-dmpna* (*DT*) across all evaluation metrics. Such poor performance may be due to the ineffectiveness of domain-invariant compound features to encode necessary task-specific information.

Surprisingly, *DANN* even performs worse than the fingerprint-based methods (i.e., *FCN-morgan* and *FCN-morganc*). As a matter of fact, overall, fingerprint-based methods perform relatively well compared to all other baselines. Compared to *GNN*-based methods, fingerprint-based methods achieve competitive or even better performance in most evaluation metrics. This could be due to potential overfitting of *GNN*-based methods in low-data settings. It is known that *GNN*s require large amounts of training data to extract relevant molecular substructures and to effectively encode meaningful task-specific information. In contrast, fingerprint-based methods are not data hungry owing to fewer learnable parameters, and thus, these methods can perform reasonably well in low-data settings.[93]

2.3. Top-*N* Task-Wise Performance Comparison. Table 3 presents a fine-grained performance comparison of top-performing methods over all 240 target tasks across different evaluation metrics. The columns corresponding to each evaluation metric have the percentage of tasks for which each method is among the top-k (k = 1, 3, 5) best methods with respect to the metric. Note that for each method we consider the best performing model that achieves the optimal performance in each evaluation metric. Therefore, for a given method, the models with the optimal performance in each metric do not necessarily have the same set of corresponding hyperparameters.

Table 3 shows that *TAc* methods achieve the top-1 best performance among more tasks compared to other methods. For example, *TAc-dmpna* is the best performing method in terms of *ROC-AUC* for 22% of tasks, that is, more than 2-fold compared to the best baseline method *FCN-morgan* (10%). *TAc-dmpna* consistently achieves the top-3 and top-5 best performance in terms of *ROC-AUC* on significantly more tasks compared to other methods, with even more folds of difference. Similar

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

Table 3. Top-*N* Performance Comparison (%)[a]

method	*ROC-AUC*			*PR-AUC*			*F1*		
top-*N*	1	3	5	1	3	5	1	3	5
FCN-morgan	10	15	20	13	21	30	11	19	23
FCN-morganc	2	13	18	3	17	22	4	17	22
FCN-dmpn	5	9	17	6	11	17	10	19	30
FCN-dmpna	4	10	26	2	7	20	5	11	26
FCN-dmpn (DT)	2	11	23	4	13	24	5	13	22
FCN-dmpna (DT)	5	18	33	6	14	31	2	8	23
DANN-dmpn	2	9	16	1	5	15	3	10	19
DANN-dmpna	2	8	17	1	8	15	2	11	19
TAc-dmpn	18	52	85	17	48	81	12	49	**81**
TAc-fc-dmpn	12	45	79	15	48	80	**17**	**50**	80
TAc-dmpna	**22**	**62**	**89**	14	55	83	13	43	78
TAc-fc-dmpna	16	51	81	**20**	**56**	**86**	15	**50**	**81**

[a]In this table, the columns *ROC-AUC*, *PR-AUC*, and *F1* have the percentage of tasks for which each method is ranked within the top-1, top-3, and top-5 best methods in respective metrics. The best performance values are in bold.

trends hold for *PR-AUC* and *F1* as the evaluation metrics. This indicates the strong performance of *TAc* methods.

Among the four *TAc* variants, *TAc-dmpna* is the best in terms of *ROC-AUC*; *TAc-fc-dmpna* is overall the best in terms of *PR-AUC* as it achieves the top-1, top-3, and top-5 best performance on more tasks compared to other methods; and *TAc-fc-dmpn* and *TAc-fc-dmpna* are the best in terms of *F1* as they are either better than or similar to other methods. This indicates that while different variants may have advantages of optimizing with respect to different evaluation metrics *TAc-fc* (Figure 7, with the feature-wise and compound-wise discriminators) is actually also a very strong method or even better compared to *TAc*.

2.4. Comparison of Discriminators. Table 4 presents a detailed performance comparison between *TAc*, *TAc-fc*, *TAc-c*, and *TAc-f*, all with *dmpna*. We use *dmpna* here because as in Table 3 *TAc-fc-dmpna* shows better performance on average compared to *TAc-fc-dmpn*. *TAc-c* and *TAc-f* are obtained by removing either the feature-wise discriminator L (Section 4.2.5.1) or the compound-wise discriminator G (Section 4.2.5.2) from *TAc-fc*. Note that for each bioassay the optimal model of each method is selected based on *ROC-AUC*. The *diff* % values in each row block are calculated as the difference (in %) of average performance in each metric from the *TAc-fc* variant over *TAc*. The *t-diff* % values in each row block are calculated as the average of task-wise performance improvements (in %) from the corresponding variant over *TAc*. The row *N-impv* in each row block denotes the number of improved target tasks where the variant performs better than *TAc* in respective metrics, and the average of task-wise performance improvement among only the improved tasks is calculated as *t-impv* % (in %).

Compared to *TAc*, *TAc-fc* achieves similar but slightly worse performance overall (i.e., −0.375% in *diff* % on *ROC-AUC*); on individual tasks, *TAc-fc* has some statistically significant worse

Table 4. Comparison on Discriminators (with *dmpna*)[a]

method	*ROC-AUC*	*PR-AUC*	*precision*	*sens*	*accuracy*	*F1*
TAc	0.801	0.786	0.731	0.729	0.720	0.713
TAc-fc	0.798	0.785	0.730	0.728	0.719	0.713
diff %	−0.375	−0.127	−0.137	−0.137	−0.139	0.000
t-diff %	−0.380	−0.119	−0.072	0.116	−0.150	−0.064
	(2.59×10^{-04})	(4.88×10^{-01})	(7.26×10^{-01})	(7.00×10^{-01})	(5.53×10^{-01})	(8.04×10^{-01})
N-impv	93 (39%)	125 (52%)	119 (50%)	111 (46%)	99 (41%)	112 (47%)
t-impv %	0.921	1.373	2.756	7.037	2.230	3.729
	(7.73×10^{-18})	(4.13×10^{-23})	(1.69×10^{-18})	(5.85×10^{-19})	(4.26×10^{-14})	(1.31×10^{-15})
TAc-c	0.801	0.786	0.730	0.734	0.721	0.716
diff %	0.000	0.000	−0.137	0.686	0.139	0.421
t-diff %	0.010	−0.080	−0.130	1.583	0.128	0.516
	(7.78×10^{-01})	(6.72×10^{-01})	(6.32×10^{-01})	(3.03×10^{-01})	(4.01×10^{-01})	(3.90×10^{-01})
N-impv	135 (56%)	119 (50%)	123 (51%)	126 (52%)	128 (53%)	130 (54%)
t-impv %	0.845	1.330	2.763	8.798	1.971	4.165
	(1.13×10^{-23})	(5.03×10^{-24})	(2.67×10^{-17})	(8.22×10^{-18})	(4.75×10^{-21})	(1.75×10^{-15})
TAc-f	0.799	0.785	0.732	0.722	0.721	0.711
diff %	−0.250	−0.127	0.137	−0.960	0.139	−0.281
t-diff %	−0.192	−0.091	0.218	−0.768	0.170	−0.326
	(4.00×10^{-02})	(3.76×10^{-01})	(5.44×10^{-01})	(1.42×10^{-01})	(3.19×10^{-01})	(4.98×10^{-01})
N-impv	100 (42%)	114 (48%)	124 (52%)	117 (49%)	125 (52%)	123 (51%)
t-impv %	1.029	1.597	2.992	6.999	2.037	3.764
	(3.22×10^{-13})	(2.22×10^{-21})	(3.85×10^{-24})	(1.11×10^{-19})	(9.65×10^{-20})	(1.12×10^{-16})

[a]In this table, the first row block has the average performance of *TAc*. Each of the other row blocks has the performance comparison of a *TAc-fc* variant with respect to *TAc*. The metric *diff* % represents the difference of average performance of each comparison method with respect to *TAc*; *t-diff* % represents the average of the task-wise improvement, with corresponding *p*-values in the parentheses below; *N-impv* represents the number of improved tasks and its proportion in the parentheses; and *t-impv* % represents the average of the task-wise improvement only among the improved tasks, with corresponding *p*-values in the parentheses below.

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

performance in terms of *ROC-AUC* but similar performance as *TAc* on other evaluation metrics. In addition, *TAc-fc* still provides significant task-wise improvement in about 40−50% of tasks across all evaluation metrics. Particularly, it improves the *ROC-AUC* score for 93 out of 240 (39%) tasks significantly by 0.921% on average (p-value = 7.73×10^{-18}) over *TAc*. This suggests that the learned feature-wise and compound-wise transferability together have the capacity of improving some target tasks. *TAc-c* performs similarly to *TAc* on average (i.e., 0.000 in *diff* % on *ROC-AUC*; no significant *t-diff* %). However, *TAc-c* improves over more than half of the tasks with statistical significance on all the evaluation metrics. For example, *TAc-c* achieves better *ROC-AUC* on 135 out of 240 (56%) tasks. This indicates that the global discriminator (Section 4.2.5.2) that differentiates compounds for the source and target tasks could help improve performance for some tasks. *TAc-f* also shows improvement on about half of the tasks (*N-impv*) with significant improvement that is even higher compared to that in *TAc-c* but with overall performance (*diff* %) still slightly worse than that of *TAc*. The fact that *TAc-fc*, *TAc-c*, and *TAc-f* improve about half of the tasks over *TAc* without discriminators indicates that they are suitable for certain tasks.

We hypothesize that *TAc-c* can effectively focus on similar compounds of source and target bioassays by learning compound-wise transferability via G. We validate this hypothesis with an additional analysis on model predictions and pairwise similarities of predicted compounds with source and target compounds. We find and study the active compounds that are correctly classified as active by *TAc-c* but incorrectly classified as inactive by *TAc* and its variants. Table S4 presents the analysis for these active compounds among target tasks which have at least one such active compound. For each of such active compounds in the target task, we calculated the mean pairwise similarities of that compound with its five most similar active compounds in the source task and in the target task, respectively.

On average, *TAc-c* correctly classifies 5.4% (i.e., average of values in "cor %" column in Table S4) of active compounds that are incorrectly classified by *TAc* and its variants. These compounds were found to be 12.4% more similar to the active compounds in the source task than to those in the target task. Furthermore, in 47 out of 97 (48%) tasks with at least one active compound only correctly classified by *TAc-c*, the similarity difference is statistically significant (p-value < 0.05). Overall, this analysis demonstrates that *TAc-c* can better learn the commonalities between source and target compounds and hence can enhance information transfer from the source task to the target task.

2.5. Parameter Study. Figure 1 presents the parameter study in *TAc-dmpna* on α (i.e., the trade-off parameter between the source and target classification losses as in eq 4). The study was conducted over the tasks for which *TAc-dmpna* outperforms the other methods. The values in each cell in the figure represent the average of the best performance over the tasks with the optimal choice of other hyperparameters.

Figure 1 shows that *TAc-dmpna* has the best average performance in *ROC-AUC*, *PR-AUC*, *precision*, *sens*, *accuracy*, and *F1*-score when α = 0.5, 0.5, 0.1, 0, 0.5, and 0.1, respectively. It indicates that weighing the source and target classification losses differently has notable effects on the overall performance. This figure also demonstrates several trends: (1) the best average performance is achieved with α = 0.1 and 0.5 (i.e., nonzero values) for all the metrics except *sens* and (2) performance degrades especially when α increases. Nonzero

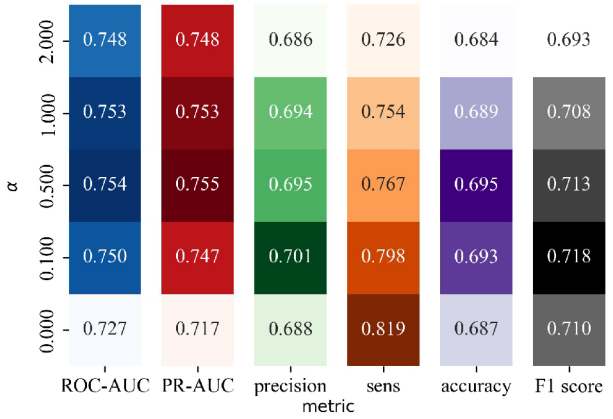


Figure 1. Parameter study of *TAc-dmpna*. The columns represent different evaluation metrics. The values in each cell have the average of the best performance achieved with given α and optimal choice of other hyperparameters. Darker cells indicate better performance.

values of α as the optimal values indicate that leveraging information from the source task is able to help improve the target task. The fact that the optimal, nonzero α values are relatively small indicates that the training is still more focused on the target tasks, while useful information is transferred from the source tasks. On the other hand, if α is too large (i.e., the source classification loss is given high weightage), the training would be dominated by the source task, and thus the trained model could not well capture the patterns in the target task. That could explain why model performance decreases when α increases.

Figure 2 presents the parameter study in *TAc-fc-dmpna* in terms of *ROC-AUC* on α (i.e., the trade-off parameter between

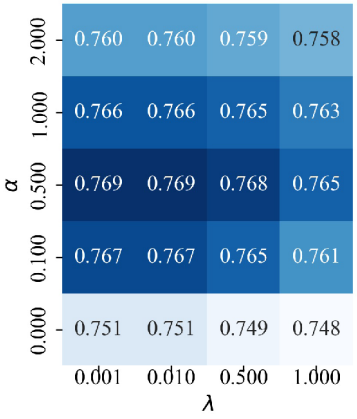


Figure 2. Parameter Study of *TAc-fc-dmpna* in terms of *ROC-AUC*.

source and target losses in eq 4) and λ (i.e., the trade-off parameter between the classification and discriminator losses in eq 13). Studies over other metrics are presented in Figure S1 in the Supporting Information. The values in each cell of this figure represent the average of the best performance over the tasks where *TAc-fc-dmpna* outperforms all other methods, with corresponding α and λ and with optimal choice of other hyperparameters.

Figure 2 shows that *TAc-fc-dmpna* has the best performance in *ROC-AUC* (i.e., 0.769) when α = 0.5 and λ = 0.01 and 0.001, that is, all nonzero values. This demonstrates that a lower weight on the source classification loss than the target classification loss and a lower weight on discriminator losses (sum of $\mathcal{L}_{(l)}$ and $\mathcal{L}_{(g)}$

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

) will enable effective transfer of relevant information from the source domain. Figure 2 also demonstrates that when α is too small or too large, regardless of what λ is, there is a significant performance drop (as indicated in the topmost rows). This effect of α can be explained following the same reasoning presented in the previous section. For the optimal α in each metric, λ = 0.01 gives the best performance for most metrics. This implies that *TAc-fc-dmpna* can effectively leverage source task data to learn transferable compound features (using L) and to selectively focus on similar compounds (using G) during training. Intuitively, for a given α, higher λ values (i.e., higher weight on discriminator losses) will encourage learning of more domain-invariant compound features. Such domain-invariant features contain little task-specific information and may not be relevant for effective activity classification for the target task, and therefore the overall performance degrades.

2.6. Case Studies: *TAc-dmpna*

2.6.1. Relation between Performance Improvement and Bioassay Similarity. Among 240 tasks, we identified and studied four tasks with a significant performance difference in *ROC-AUC* from *TAc-dmpna* over the best no-transfer baseline method (i.e., *FCN-dmpna*). Figure 3 presents the average pairwise similarity

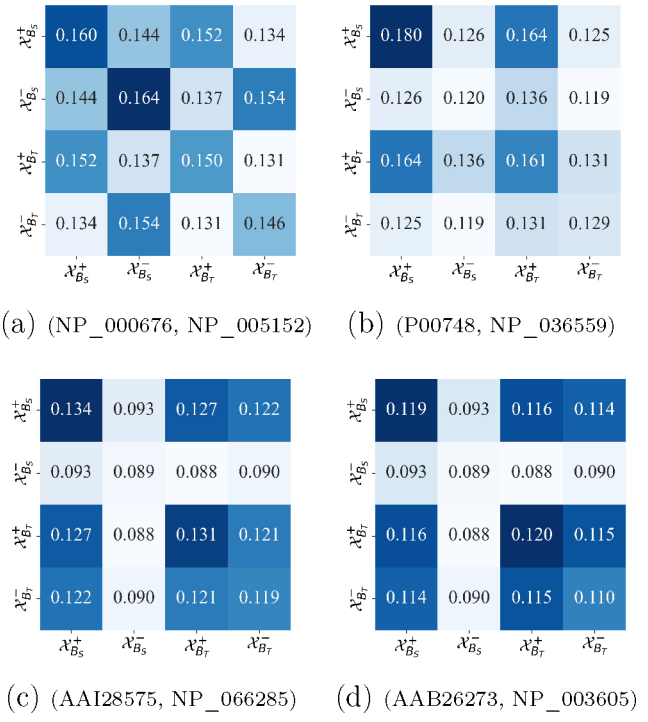


Figure 3. Similarity matrices of target pairs with significant *ROC-AUC* improvement/degradation.

matrices of the four task pairs (captions include the corresponding bioassay PubChem AIDs of the source bioassay and the target bioassay), where Figure 3a and 3b have the target tasks that are significantly improved by *TAc-dmpna* and Figure 3c and 3d have the target tasks that are significantly degraded. In the figure, $\mathcal{X}^+_{B_S}$, $\mathcal{X}^-_{B_S}$, $\mathcal{X}^+_{B_T}$ and $\mathcal{X}^-_{B_T}$ denote the active ($^+$) and inactive ($^-$) compounds for the source (B_S) and target (B_T) tasks, and average compound similarities (sim) were calculated using the Tanimoto coefficient over Morgan-count fingerprints (with radius = 3 and dimension = 2048).

In Figure 3a and 3b, the performance of the target task NP_005152 and NP_036559 was improved from *TAc-dmpna* over *FCN-dmpna* by 34.13% and 27.10%, respectively. Figure 3a and 3b show that for these two target tasks sim($\mathcal{X}^+_{B_S}$, $\mathcal{X}^+_{B_T}$) (0.152 in Figure 3a, 0.164 in Figure 3b) is notably greater than both sim($\mathcal{X}^+_{B_S}$, $\mathcal{X}^-_{B_T}$) (0.134 in Figure 3a, 0.125 in Figure 3b) and sim($\mathcal{X}^-_{B_S}$, $\mathcal{X}^+_{B_T}$) (0.137 in Figure 3a, 0.136 in Figure 3b). This indicates that if active compounds across bioassays are more similar than compounds with different activity labels across bioassays *TAc-dmpna* can better capture the commonalities among those similar active compounds and can better transfer relevant information across bioassays. This transferred information can effectively improve the target task performance. On the other hand, if compounds with different activity labels across bioassays are more similar than compounds with the same activity labels, *TAc-dmpna* can cause transfer of conflicting information. Such a transfer can result in performance degradation for the target task. Such performance degradation in *ROC-AUC* from *TAc-dmpna* over *FCN-dmpna* for the target tasks in pairs (AAI28575, NP_066285) in Figure 3c was 5.74% and (AAB26273, NP_003605) in Figure 3d was 2.34%, respectively. In Figure 3c and 3d, sim($\mathcal{X}^+_{B_S}$, $\mathcal{X}^+_{B_T}$) and sim($\mathcal{X}^+_{B_S}$, $\mathcal{X}^-_{B_T}$) values are relatively similar (0.127 vs 0.122 in Figure 3c, 0.116 vs 0.114 in Figure 3d). This indicates that when the similarities between $\mathcal{X}^+_{B_S}$ and $\mathcal{X}^-_{B_T}$ compounds are relatively high *TAc-dmpna* could lead to transfer of conflicting information, causing inactive compounds in the target bioassay to be incorrectly classified as active.

Furthermore, we analyzed the relation between the task-wise *ROC-AUC* improvement from *TAc* over *FCN-dmpna* and the bioassay similarities. Figure 4 presents such a relation. Note that

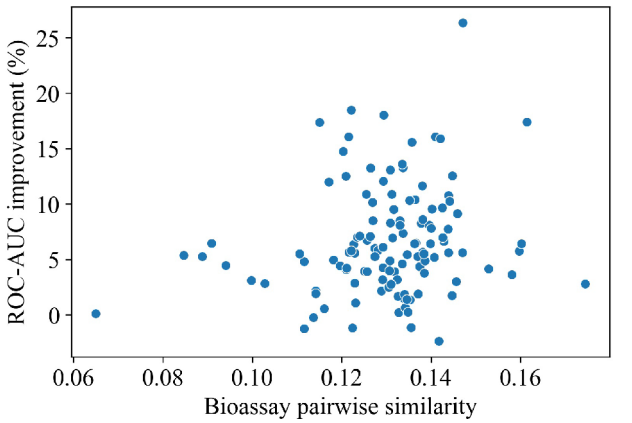
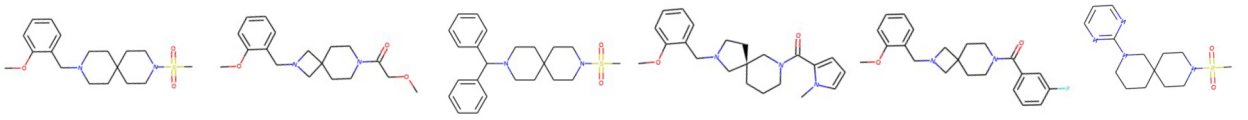


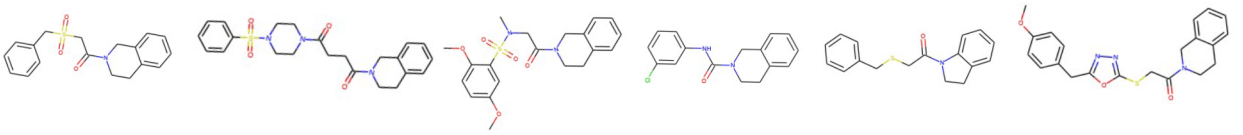
Figure 4. *ROC-AUC* improvement from *TAc* over *FCN-dmpna* vs bioassay similarity.

the bioassay similarities are calculated as the average of all pair-wise compound similarities across two bioassays in the same way as discussed in Section 2.1.1.2. Figure 4 demonstrates that there are significant task-wise *ROC-AUC* improvements (e.g., in the upper right region) when the pair-wise similarities are relatively high (e.g., greater than 0.12), and there are marginal improvements (e.g., in the lower left region) when the pair-wise similarities are low (e.g., lower than 0.12). This suggests that if bioassay pairs are more similar *TAc* can improve the performance over *FCN-dmpna* by a large margin (e.g., more than 10%). On the other hand, if bioassay pairs are less similar, *TAc*

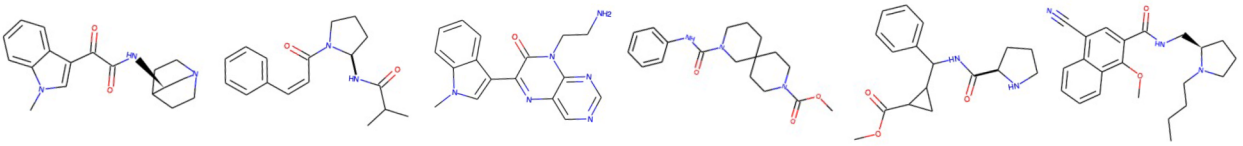
https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483



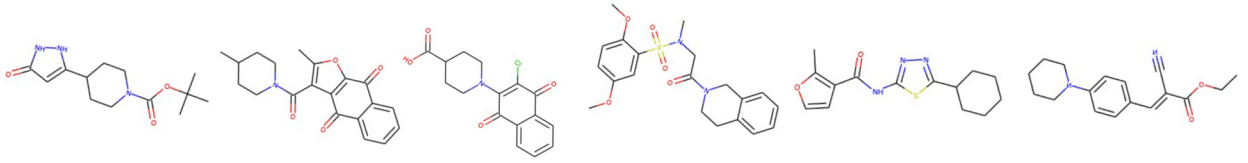
(a) A *correctly* classified compound from the target bioassay NP_000752 and its top-5 most similar compounds from the source bioassay NP_000762



(b) A *correctly* classified compound from the target bioassay NP_660205 and its top-5 most similar compounds from the source bioassay NP_004337



(c) An *incorrectly* classified compound from the target bioassay NP_000752 and its top-5 most similar compounds from the source bioassay NP_000762



(d) An *incorrectly* classified compound from the target bioassay NP_660205 and its top-5 most similar compounds from the source bioassay NP_004337

Figure 5. Visualization of a few selected compounds from the target bioassay and their corresponding top 5 most similar compounds from the source bioassay.

achieves little or no improvement over *FCN-dmpna*. Indeed, there are some bioassay pairs that are more similar, yet *TAc* achieves marginal or negative improvement (e.g., in the lower middle region). This is possibly due to the fact that the performance improvement is not only a function of bioassay similarity; in fact, the improvement can be marginal or negative owing to poor generalization during testing.

2.6.2. Correctly Classified Compounds Possibly Due to More Similar Compounds in the Source Bioassay. In this section, we identified (i) a few compounds that were correctly classified by *TAc* but incorrectly classified by the baselines and (ii) a few compounds that were incorrectly classified by *TAc* but correctly classified by the baselines. Figure 5(a) and (b) presents two such examples for (i), and Figure 5(c) and (d) presents two such examples for (ii). In each figure, the left-most compound is the compound to be classified as active/inactive from the target bioassay (referred to as $\mathbf{x}^{(T)}$), and the others are the top 5 most similar compounds (referred to as $\mathcal{X}^*_{B_S}$) to $\mathbf{x}^{(T)}$ from the corresponding source bioassay. The mean pairwise Tanimoto coefficients between $\mathbf{x}^{(T)}$ and $\mathcal{X}^*_{B_S}$ in Figure 5(a), (b), (c), and (d) are 0.407, 0.428, 0.210, and 0.143, respectively. Thus, in Figure 5(a) and (b), $\mathbf{x}^{(T)}$ values are structurally more similar to their corresponding $\mathcal{X}^*_{B_S}$. Relatively, in Figure 5(c) and (d), $\mathbf{x}^{(T)}$ values are less similar to their corresponding $\mathcal{X}^*_{B_S}$. This suggests that *TAc* classifies some compounds correctly, probably due to the fact that those compounds have very similar compounds in the source bioassay.

2.7. Compound Prioritization Using *dmpna*. We also explored the potential of using *dmpna* for compound prioritization purposes. We developed a comprehensive learning-to-rank method *gnnCP* for effective compound prioritization that jointly learns molecular graph representations via *GNN* and a scoring function using the representations. The learning methods for compound prioritization are described in Section 4 in the Supporting Information.

2.7.1. Materials. Baselines. We compare *gnnCP* with the following feature vectors using the same scoring and loss functions: (i) binary Morgan fingerprints (*morgan*), (ii) morgan count fingerprints, (*morgan-c*), (iii) bioassay-specific compound features[12] computed using the Tanimoto coefficient on binary Morgan fingerprints (*morgan-ba*), (iv) 200-dimensional RDKit descriptors (*RDKit200*), and (v) directed message passing network[19] (*dmpn*). We generate binary Morgan fingerprints and Morgan-count fingerprints (with radius = 2 and size = 2048) using RDKit.[90] Codes for computing the RDKit descriptors are available in the Descriptastorus package.[94]

Experimental Protocol. In order to evaluate the overall ranking performance, we perform a 5-fold cross validation. We randomly split each bioassay into five folds. In each run, four folds of each bioassay are used for training, and the other fold is used for testing. We record optimal values of each performance

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

Table 5. Overall Performance Comparison of *gnnCP*

method	CI	R@3	R@5	ndcg@3	ndcg@5	R@5%	ndcg@5%
morgan	0.706	0.543	0.644	0.814	0.816	0.420	0.838
morgan-c	0.711	0.545	0.655	0.815	0.819	0.437	0.846
morgan-ba	0.687	0.500	0.626	0.789	0.797	0.375	0.816
RDKit200	0.687	0.519	0.632	0.790	0.797	0.396	0.813
dmpn	0.731	0.643	0.709	0.854	0.847	0.579	0.896
dmpna	**0.748**	**0.686**	**0.740**	**0.881**	**0.867**	**0.686**	**0.936**
diff %	2.353	6.608	4.460	3.114	2.421	18.428	4.475
t-diff %	2.535	7.645	4.720	3.406	2.569	24.578	4.979
p-value	1.14e-10	4.89e-10	9.87e-13	1.42e-12	2.25e-12	3.95e-15	1.83e-11

In this table, the columns have the respective average of each performance metric over all bioassays obtained by the respective optimal hyperparameter settings. The best/second best performance under each metric is bold/underlined.

metric averaged over the five folds. Finally, we report the average of all such recorded optimal values of each performance metric over all the bioassays. For each bioassay, we train the models using an Adam[92] optimizer with an initial learning rate $\in \{5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}\}$. We use grid search to tune all the hyperparameters such as the dimension of the graph representation d, hidden dimension of the attention layer, and batch size. Specifically, we use $d \in \{25, 50, 100\}$ for *dmpn* and *dmpna* and a hidden dimension of the attention layer $\in \{5, 10, 20\}$ for *dmpna*. We use batch size $\in \{128, 256, 512\}$ and $\lambda = 1 \times 10^{-6}$ for all the models. All the models are trained for 50 epochs.

Evaluation. We evaluate all the methods using a set of 105 single-target confirmatory bioassays from PubChem.[20] These bioassays all use IC_{50} to measure compound binding affinities and have at least 50 active compounds. For each bioassay, we only keep the active compounds and remove duplicate compounds and those with identical IC_{50} values. We evaluate the ranking performance using concordance index (CI), recall@k (R@k), and normalized discounted cumulative gain@k (ndcg@k),[12] where k = 3, 5, and 10. We also use R@k% and ndcg@k%, where we consider the top k% (k = 5, 10) of the test fold compounds in r.

Table 5 presents the performance comparison between *dmpna*, *dmpn*, and the baselines. Overall, *dmpna* significantly performs better than all the baselines including *dmpn*, across all performance metrics. The average performance improvement from *dmpna* over *dmpn* in terms of CI, recall@3, recall@5, ndcg@3, ndcg@5, recall@5%, and ndcg@5% is 2.353%, 6.608%, 4.460%, 3.114%, 2.421%, 18.429%, and 4.475%, respectively. Furthermore, compared to *dmpn*, the average bioassay-wise performance improvement from *dmpna* is most significant in terms of recall@3, recall@5, ndcg@3, ndcg@5, recall@5%, and ndcg@5% (p-values: 4.89×10^{-10}, 9.87×10^{-13}, 1.42×10^{-12}, 2.25×10^{-12}, 3.95×10^{-15}, and 1.83×10^{-11}, respectively). This indicates that *dmpna* can rank the topmost compounds better than *dmpn*. Unlike mean pooling in *dmpn*, the attention mechanism in *dmpna* can differentially focus on atoms based on the relevance of each atom to the prioritization problem. This demonstrates the ability of *dmpna* to better differentiate compounds and to achieve effective compound prioritization. Furthermore, *dmpna* and *dmpn* significantly outperform all the fingeprint-based baselines across all performance metrics. Compared with the best performing fingerprint-based baseline *morgan-c*, in terms of CI, recall@3, recall@5, ndcg@3, ndcg@5, recall@5%, and ndcg@5%, the average performance improvement from *dmpna* is 5.247%, 25.724%, 13.094%, 8.140%, 5.871%, 56.875%, and 10.637%, respectively, and from *dmpn* in terms of CI, recall@3, recall@5, ndcg@3, ndcg@5, recall@5%, and ndcg@5% is 2.827%, 17.932%, 8.266%, 4.874%, 3.369%, 32.464%, and 5.898%, respectively. This demonstrates that the learned representation out of *gnnCP* can effectively encode useful molecular substructure information and thus is more effective for compound prioritization.

3. CONCLUSIONS

We have developed *TAc* that effectively leverages source bioassay data to improve the performance of the target task. We also proposed a variant of *TAc*, i.e., *TAc-fc*, that additionally learns feature-wise and compound-wise transferability. We conducted an exhaustive array of experiments and analyses that suggest that *TAc-dmpna* is the best-performing method on average across all target tasks. The proposed variant is also a very strong method and even better compared to *TAc* on certain target tasks. Furthermore, in ablation studies, we also showed that *TAc-fc-dmpna* can even improve performance for more than half of the target tasks compared to *TAc-dmpna*. Our analyses further demonstrated that learning compound-wise transferability via G can better encode the commonalities between compounds across bioassays. We also provided a parameter study to demonstrate the effect of α and λ on our proposed methods. Additionally, we demonstrated the efficacy of our proposed *dmpna* in both compound activity and compound prioritization problems since it performed better than any other compound representation methods.

In this work, we paired the bioassays if their corresponding protein targets belong to the same protein family. In other words, when we paired the bioassays, the corresponding pair of tasks is assumed to be related. We assumed that leveraging activity information from related protein targets (i.e., targets belonging to the same protein family) can improve the target task performance. However, we observed that *TAc* did not improve all the targets compared to the best no-transfer baseline method *FCN-dmpna*. This suggests the occurrence of potential negative transfer. In future works, we will focus on developing a more principled approach to determine task-relatedness. Given a target task, our current method only considers a single source task. This severely limits the scope of transfer from only one related task and can also impact the performance on the target task if the learning is too focused on the source task. Our future work will incorporate multiple source tasks for each target task by simultaneously learning task-relatedness in a data-driven manner.

4. COMPUTATIONAL METHODS

4.1. Notations and Definitions. In this section, we listed the notations and definitions used in this paper. Table 6 presents

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

Table 6. Notations

method	meanings
c/B	compound/bioassay
$\mathcal{G} = (\mathcal{A}, \mathcal{E})$	molecular graph with set of atoms $\mathcal{A}$ and bonds $\mathcal{E}$
u	an atom in $\mathcal{G}$
(u, v)	a bond connecting atoms u and v in $\mathcal{G}$
$\mathcal{N}$(u)	neighbors of atom u in $\mathcal{G}$
$\mathcal{X}$	set of compounds in a bioassay
$\mathcal{Y}$	set of labels corresponding to $\mathcal{X}$
$\mathbb{X}$	input feature space
$\mathbb{Y}$	label space
$\mathcal{D} = \{\mathbb{X}, P(\mathcal{X})\}$	a domain consisting of $\mathbb{X}$ and marginal probability distribution $P(\mathcal{X})$
$\mathcal{T} = \{\mathbb{Y}, \omega(\cdot)\}$ $T = \{Y,\omega(\cdot)\}$	a task consisting of label space and a decision function $\omega(\cdot)$
h	hidden state
r	molecular representation out of *GNN*
z	scaled molecular representation

a list of notations and their meanings. We represent a compound and using a molecular graph, $\mathcal{G}_c$. $\mathcal{G}_c$ is denoted as $\mathcal{G}_c = (\mathcal{A}_c, \mathcal{E}_c)$, where $\mathcal{A}_c$ is the set of atoms and $\mathcal{E}_c$ is the set of corresponding bonds in c. We denote the set of compounds in a bioassay B as $\mathcal{X}_B$ and the activity labels of those compounds accordingly as $\mathcal{Y}_B$. In this paper, we use a label "1" or "0" to indicate that a compound is active or inactive in a bioassay, respectively.

We use the following definitions related to transfer learning.

Domain: a domain $\mathcal{D}$ is a set of labeled compounds $\mathcal{D} = \{c_i = (\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathbb{X}, y_i \in \mathbb{Y}, i = 1, ..., |\mathcal{D}|\}$, where the compounds $\{\mathbf{x}_i\}$ are represented in a feature space $\mathbb{X}$, and their activity labels $\{y_i\}$ are represented in a label space $\mathbb{Y}$; $|\mathcal{D}|$ is the size of the domain (i.e., the number of $(\mathbf{x}_i, y_i)$ pairs). In our transfer learning, we will have two domains: a source domain, denoted as $\mathcal{D}^{(S)}$, and a target domain, denoted as $\mathcal{D}^{(T)}$. In general, these two domains can have different numbers of compounds with different compound feature representations and also different label sets. We use superscript $^{(S)}$ and $^{(T)}$ to represent information associated with the source domain and the target domain, respectively. For example, $\mathbf{x}^{(T)}$ represents a compound from the target domain. In addition, we use $\mathcal{X}$ to represent the set of compound features $\{\mathbf{x}_i\}$, that is, $\mathcal{X} = \{\mathbf{x}_i|\mathbf{x}_i \in \mathbb{X}\}$, and $\mathcal{Y}$ to represent the set of compound labels, that is, $\mathcal{Y} = \{y_i|y_i \in \mathbb{Y}\}$. Thus, $\mathcal{D}$ can also be represented as $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$.

Task: Given a domain $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1,...,|\mathcal{D}|}\}$, a task $\mathcal{T}$ is to learn a model that maps each $\mathbf{x}_i$ to its corresponding y_i. In our transfer learning, we will have two tasks: a source task, denoted as $\mathcal{T}^{(S)}$, and a target task, denoted as $\mathcal{T}^{(T)}$. $\mathcal{T}^{(S)}$ and $\mathcal{T}^{(T)}$ learn from the source domain $\mathcal{D}^{(S)}$ and the target domain $\mathcal{D}^{(T)}$, respectively.

Transfer Learning: Transfer learning learns and transfers information from the source task $\mathcal{T}^{(S)}$ to the target task $\mathcal{T}^{(T)}$ and helps improve the performance of $\mathcal{T}^{(T)}$. The underlying assumptions are that: (1) the target domain $\mathcal{D}^{(T)}$ does not have sufficient information for $\mathcal{T}^{(T)}$ to learn a good model, and (2) there are commonalities between $\mathcal{D}^{(S)}$ and $\mathcal{D}^{(T)}$; such commonalities can be transferred from $\mathcal{D}^{(S)}$ to $\mathcal{D}^{(T)}$ and used to improve $\mathcal{T}^{(T)}$.

4.2. Methods. In this section, we present our two transfer learning methods: *TAc* and *TAc-fc*. We first introduce the overall architecture of *TAc* in Section 4.2.1 and then discuss each component in detail in subsequent sections (i.e., Sections 4.2.2 and 4.2.3). We discuss the end-to-end optimization process in Section 4.2.4. We then introduce *TAc-fc* with additional components that learn feature-wise and compound-wise transferability and finally discuss the optimization process in Section 4.2.5.

4.2.1. Overall Architecture of TAc. *TAc* learns to generate transferable features that can generalize well from one domain to another and increases the predictive power for classification in the target domain. Figure 6 presents the overall architecture of the proposed *TAc*. *TAc* consists of two components: (1) a feature learner F that learns to represent chemical compounds and (2) a domain-wise classifier *S* that classifies chemical

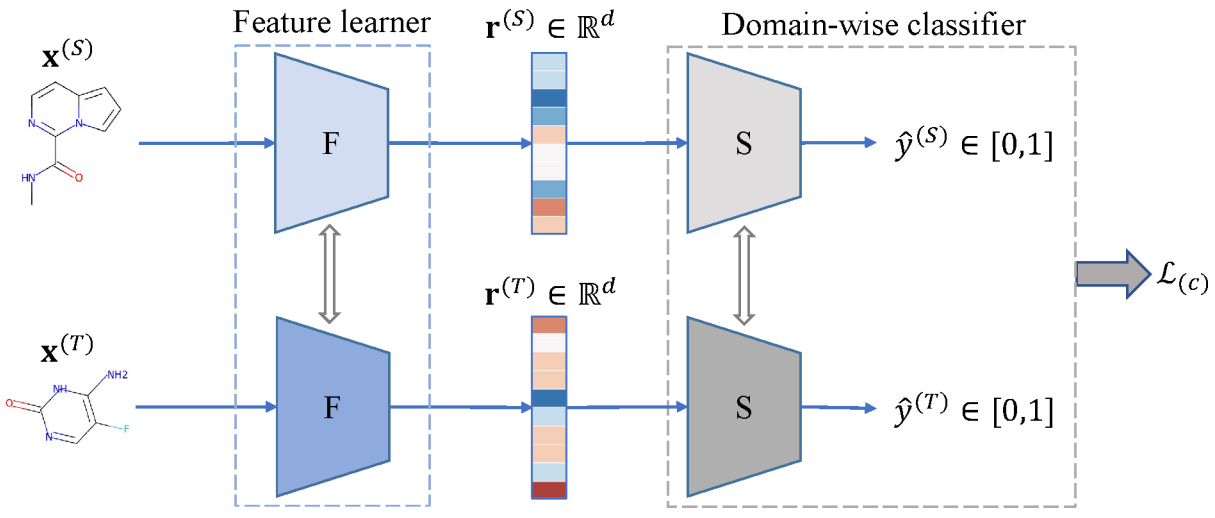


Figure 6. Proposed architecture of *TAc*. The feature learner F learns compound representations **r** given the corresponding molecular graph. The domain-wise classifier S classifies the compound as active/inactive.

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

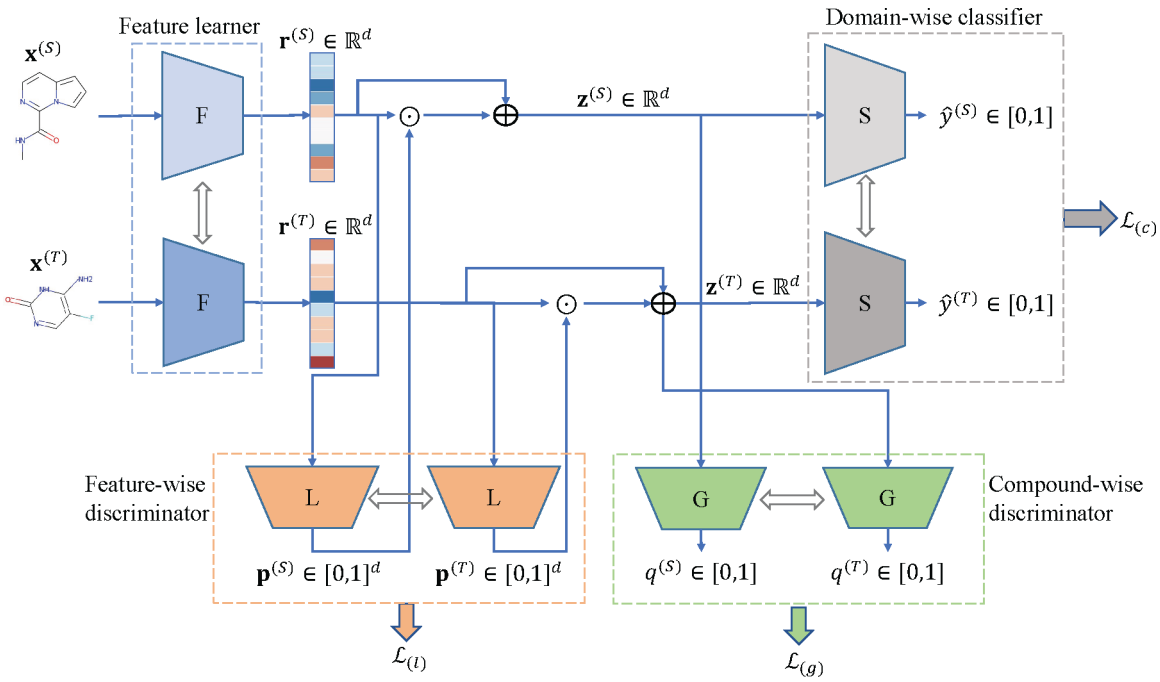


Figure 7. Proposed architecture of *TAc-fc*. The feature learner F learns compound embedding **r** given the corresponding molecular graph. The feature-wise discriminator *L* learns feature-wise transferability given the learned compound embedding **r**. **r** is further scaled into **z** using its feature entropy from **p** out of *L*. The compound-wise discriminator *G* learns the compound-wise transferability given **z**. The domain-wise classifier S classifies the compound as active/inactive.

compounds of each domain. Below, we discuss each component of *TAc* in detail.

4.2.2. Learning Compound Representations. This section describes how the feature representations of compounds are learned. In *TAc*, the feature representations of chemical compounds are learned in a data-driven fashion. Compared to using static fingerprints or fixed feature representations of molecular structures,[33] such learned features will be more adapted to the learning task and enable optimal performance. We leverage the popular idea of graph neural networks[95] and use the Directed Message Passing Neural Network, denoted as *dmpn*, developed in Yang et al.[19] Given a molecular graph $\mathcal{G}_c = (\mathcal{A}_c, \mathcal{E}_c)$ for a compound c, *dmpn* learns a feature vector, also called an embedding of c using graph convolution, by passing messages along directed edges over molecular graphs. In *dmpn*, two representations for each bond are learned via message passing through the two directions along the bond. Then atom representations are learned from the representations of their incoming bonds. In the end, the compound representation is generated via mean pooling over all the atom representations. Details about *dmpn* are presented in Section 1 in the Supporting Information and are also available in Yang et al.[19]

Based on *dmpn*, we further improve the compound representation learning by introducing an attention mechanism inspired from Graph Attention Networks.[96] This new method is referred to as *dmpn* with attention, denoted as *dmpna*. Specifically, we replace the mean pooling in *dmpn* with an attention-based pooling mechanism as follows

$$\boldsymbol{r}_c = \sum_{u \in \mathcal{A}_c} (1 + w_u) \odot \boldsymbol{s}_u \tag{1}$$

where $\odot$ is the element-wise product; $\mathbf{s}_u$ is the learned representation of atom u as in *dmpn*; and w_u is the attention weight on atom u calculated as follows

$$w_u = \frac{\exp(f_a(\mathbf{s}_u))}{\sum_{v \in \mathcal{A}_c} \exp(f_a(\mathbf{s}_v))} \tag{2}$$

where $f_a(\cdot)$ is a 2-layer feed-forward network with a ReLU activation function after the hidden layer. That is, the attention learns a specific weight on each atom. Thus, the attention mechanism in *dmpna* can differentially focus on atoms based on the relevance of each atom toward the final predictive task. The network to learn compound embeddings is denoted as *F* (i.e., *F* is *dmpn* or *dmpna*).

4.2.3. Learning to Classify Compounds of Each Domain. This section describes how the compounds of each domain are classified as active/inactive using the learned feature representations. Given the compound embedding **r**, the domain-wise classifier classifies each compound in a given domain as active or inactive with respect to that domain using a two-layer fully connected neural network *S* as follows

$$\hat{y} = \mathrm{S}(\boldsymbol{r}) \tag{3}$$

with ReLU at the hidden layer and sigmoid at the output layer. The outputs of *S* are the probabilities of input compounds from the source/target domain being active in the source/target domain.

To learn *S*, the loss function $\mathcal{L}_{(c)}$ for the classifier is defined as follows

$$\begin{aligned}\mathcal{L}_{(c)}(\Omega, \Phi) = -\alpha \frac{1}{n^{(S)}} \sum_{\mathbf{x}^{(S)} \in \mathcal{X}^{(S)}} [y^{(S)} \log(\hat{y}^{(S)}) \\ + (1 - y^{(S)}) \log(1 - \hat{y}^{(S)})] - \frac{1}{n^{(T)}} \sum_{\mathbf{x}^{(T)} \in \mathcal{X}^{(T)}} [y^{(T)} \log(\hat{y}^{(T)}) \\ + (1 - y^{(T)}) \log(1 - \hat{y}^{(T)})]\end{aligned} \tag{4}$$

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

where $y^{(S)}/y^{(T)}$ is the ground-truth activity label of each compound in domain S/T; $n^{(S)}/n^{(T)}$ is the number of compounds in $\mathcal{X}^{(S)}/\mathcal{X}^{(T)}$ (i.e., $n = |\mathcal{D}|$); α is a hyperparameter to trade-off the two classification losses; and Ω and Φ are learnable parameters of F and S, respectively. Please note that both the source domain and the target domain use the same classifier *S*. Therefore, if the source and target domain have common compounds or very similar compounds, when these compounds have the same labels in the two domains, they will induce small classification errors in both domains; when these compounds have different labels in the two domains, they will induce large errors in one domain and small errors in the other. By minimizing the loss $\mathcal{L}_{(c)}(\Omega, \Phi)$, it will encourage common or similar compounds that have the same labels in the two domains to be more focused on through learning and prevent the transfer of conflicting information across domains.

4.2.4. TAc Model Optimization. This section presents the optimization process of the proposed *TAc*. *TAc* constructs an end-to-end transfer learning framework with the above two components: (1) feature learner F and (2) domain-wise classifier S. We solve for *TAc* through minimizing the loss function $\mathcal{L}_{(c)}$. In other words, we solve the following optimization problem

$$\hat{\Omega}, \hat{\Phi} = \underset{\Omega,\Phi}{\operatorname{argmin}} \mathcal{L}_{(c)}(\Omega, \Phi) \tag{5}$$

where Ω and Φ are the learnable parameters of F and S, respectively. Minimizing $\mathcal{L}_{(c)}$ will minimize the classification error in each domain while preventing transfer of conflicting information across domains, hence enabling the feature learner F to learn better compound features for effective classification in each domain. Since the same F and S are used for both the source and target tasks, minimizing $\mathcal{L}_{(c)}$ also enables transfer of relevant information through the shared parameters of F and S. Intuitively, the amount of transferable information from the source domain to the target domain is determined by the degree of task relatedness between those domains. In this work, the degree of task relatedness between the source and target domains is essentially controlled through the hyperparameter α. We will consider learning task relatedness or α in a data-driven manner in our future works.

4.2.5. Variant of TAc: TAc-fc. In this section, we propose a variant of *TAc* where we incorporate additional components to selectively learn feature-wise and compound-wise transferability. We denote this variant as *TAc-fc*. Figure 7 presents the overall architecture of the proposed *TAc-fc*. In addition to the feature learner F and the label classifier S, *TAc-fc* consists of two more components: (1) a feature-wise discriminator *L* that learns the transferability of each learned feature (Section 4.2.5.1) and (2) a compound-wise discriminator *G* that separates chemical compounds into source and target domains (Section 4.2.5.2). We refer to *TAc* with the feature-wise discriminator only as *TAc-f* and *TAc* with the compound-wise discriminator only as *TAc-c*. Below, we discuss each component in detail and also the optimization of the proposed method.

4.2.5.1. Learning Transferability of Individual Features. Given the learned compound embedding $\mathbf{r} \in \mathbb{R}^d$ out of F (discussed in Section 4.2.2), the feature-wise discriminator of *TAc-fc* learns the transferability of each embedding feature in $\mathbf{r}$ using a two-layer neural network *L* as follows

$$\mathbf{p} = L(\mathbf{r}) \tag{6}$$

where *L* has a hidden layer with ReLU and an output layer with sigmoid. Note that $\mathbf{p} = [p_1, p_2, ..., p_d]$ has the same dimension as $\mathbf{r}$, and $p_i \in [0, 1]$ represents the probability that the *i*-th embedding feature in $\mathbf{r}$ is specific to the source domain. Thus, the feature-wise discriminator determines whether the input compound features (not the input compounds) belong to the source domain or not. For bioactivity prediction problems, if $\mathcal{D}^{(S)}$ and $\mathcal{D}^{(T)}$ have compounds for protein targets that are from the same protein family, it is very likely that their active compounds are similar and share similar substructures (e.g., pharmacophores). In this case, intuitively, the feature-wise discriminator here could learn and represent such similar substructures.

We further quantify the transferability of each embedding feature using its entropies as follows

$$H(p_i) = -p_i \log p_i - (1 - p_i)\log(1 - p_i) \tag{7}$$

If p_i is very large or very small and has a low entropy, it indicates the *i*-th embedding feature is very likely or very unlikely to be specific to the source domain, and thus it is less likely to be common across domains; if p_i is close to 0.5 and with a high entropy, the feature is less specific to any of the domains and more likely to be common across domains and therefore can be used for information transfer across domains.

We then scale compound embedding $\mathbf{r}$ into $\mathbf{z}$ using feature entropies as follows

$$\mathbf{z} = (1 + \mathbf{H}) \odot \mathbf{r} \tag{8}$$

where $\mathbf{H} = [H(p_1), H(p_2), ..., H(p_d)] \in \mathbb{R}^d$ and $\odot$ represents element-wise dot product. Each feature is scaled with its entropy and added with itself. Intuitively, the self-addition reduces the loss of informative features due to improper scaling. Thus, in $\mathbf{z}$, domain-invariant embedding features are scaled larger than domain-specific embedding features ($\mathbf{H} \geq 0$). We will use $\mathbf{z}$ as input to the following components.

To learn the feature-wise discriminator, the loss function $\mathcal{L}_{(l)}$ is defined as follows

$$\mathcal{L}_{(l)}(\Omega, \Theta) = -\frac{1}{n^{(S)}} \sum_{\mathbf{x}^{(S)} \in \mathcal{X}^{(S)}} \frac{1}{d} \sum_{i=1 \dots d} \log(p_i^{(S)}) - \frac{1}{n^{(T)}} \sum_{\mathbf{x}^{(T)} \in \mathcal{X}^{(T)}} \frac{1}{d} \sum_{i=1 \dots d} \log(1 - p_i^{(T)}) \tag{9}$$

where $n^{(S)}/n^{(T)}$ is the number of compounds in $\mathcal{X}^{(S)}/\mathcal{X}^{(T)}$ (i.e., $n = |\mathcal{D}|$); Ω and Θ are learnable parameters of F (compound representation learning network as in Section 4.2.2) and *L* (feature-wise discriminator network as in eq 6), respectively, and *d* is the dimension of compound feature embeddings. Note that in eq 9 $p_i^{(S)}$ and $p_i^{(T)}$ both measure an embedding feature's probability of being specific to the *source domain*; superscripts $^{(S)}/^{(T)}$ here indicate that the compounds, whose features are measured, are from the source/target domain, respectively.

To have an accurate feature-wise discriminator, embedding features specific to the source/target domain should have large/small probabilities (i.e., large $p_i^{(S)}$ and small $p_i^{(T)}$) with respect to the source domain and thus make the $\mathcal{L}_{(l)}$ value small. Therefore, minimizing $\mathcal{L}_{(l)}$ will encourage accurate probabilities. Meanwhile, the feature learner F should encourage the learning of more transferable embedding features, which will have probabilities close to 0.5 and thus make the $\mathcal{L}_{(l)}$ value large.

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

Therefore, maximizing $\mathcal{L}_{(l)}$ will encourage more transferable embedding features being learned and learned well. To combine these two aspects, an adversarial optimization will be applied to $\mathcal{L}_{(l)}$ as will be described later in Section 4.2.5.3.

4.2.5.2. Learning Transferability of Compounds. Inspired by the principle that similar compounds tend to bind to similar protein targets, our method identifies such similar compounds that have the same activity labels across two targets and hence learns compound-wise transferability. Given the scaled compound embedding $\mathbf{z}$ of compound c, the compound-wise discriminator classifies whether the compound is from the source domain using a two-layer fully connected neural network G as follows

$$q = G(\mathbf{z}) \tag{10}$$

with ReLU at the hidden layer and the sigmoid at the output layer. If q is very large or very small, c is very likely or very unlikely to belong to the source domain (it is equivalent to calculating the value with respect to the target domain since there are only two domains to consider). If q is close to 0.5, c is likely to be common across domains (e.g., identical or similar compounds in the two domains) and thus can be used for information transfer across domains.

To learn the compound-wise discriminator, the loss function $\mathcal{L}_{(g)}$ is defined as follows

$$\mathcal{L}_{(g)}(\Omega, \Psi) = -\frac{1}{n^{(S)}} \sum_{\mathbf{x}^{(S)} \in \mathcal{X}^{(S)}} \log(q^{(S)}) - \frac{1}{n^{(T)}} \sum_{\mathbf{x}^{(T)} \in \mathcal{X}^{(T)}} \log (1 - q^{(T)}) \tag{11}$$

where $n^{(S)}/n^{(T)}$ is the number of compounds in $\mathcal{X}^{(S)}/\mathcal{X}^{(T)}$ (i.e., $n = |\mathcal{D}|$); Ω and Ψ are learnable parameters of F (compound representation learning network as in Section 4.2.2) and L (eq 10); and d is the dimension of the compound feature embeddings. Note that in eq 11 $q^{(S)}$ and $q^{(T)}$ represent the probability of $c^{(S)}$ and $c^{(T)}$ belonging to the *source domain*. Also, all the compounds from the source and target domains will be predicted using the same G.

In order to identify similar compounds across domains, the discriminator needs to identify compounds with their q values close to 0.5; when the q values are close 0.5, $\mathcal{L}_{(g)}$ will be maximized. Therefore, maximizing $\mathcal{L}_{(g)}$ will encourage more transferable compounds being learned and learned well. Meanwhile, to have an accurate compound-wise discriminator, compounds specific to the source/target domain should have large/small probabilities (i.e., large $q_i^{(S)}$ and small $q_i^{(T)}$) with respect to the source domain and thus make the $\mathcal{L}_{(g)}$ value small. Therefore, minimizing $\mathcal{L}_{(g)}$ will encourage accurate probabilities. To combine these two aspects, similarly as to $\mathcal{L}_{(l)}$, an adversarial optimization will be applied to $\mathcal{L}_{(g)}$ as will be described later in Section 4.2.5.3.

According to G, a compound that is common in the two domains or is similar to compounds in the other domain could be transferable (q value close to 0.5; not specific to the source or target domain). However, such common or similar compounds may have different activity labels in the two domains. Using transferred information from common/similar compounds with conflicting labels in $\mathcal{T}^{(T)}$ will confuse any learners adversely. The compound-wise discriminator G does not consider activity label information in learning compound transferability and thus possibly induces conflicting information into $\mathcal{T}^{(T)}$. However, in the downstream domain-wise classification (Section 4.2.3), the minimization of domain-specific classification errors will prevent the transfer of conflicting information.

However, the input to the domain-wise classifier S in *TAc-fc* is $\mathbf{z}$ instead of $\mathbf{r}$ as in Section 4.2.3. Given the scaled compound embedding $\mathbf{z}$, the domain-wise classifier classifies each compound in a given domain as active or inactive with respect to that domain using a two-layer fully connected neural network S as follows

$$\hat{y} = S(\mathbf{z}) \tag{12}$$

with ReLU at the hidden layer and sigmoid at the output layer. As discussed in Section 4.2.3, minimizing the loss $\mathcal{L}_{(c)}(\Omega, \Phi)$ enables correct classification in each domain and prevents the transfer of conflicting information across domains.

4.2.5.3. TAc-fc Model Optimization. This section presents the optimization process of the proposed *TAc-fc*. *TAc-fc* constructs an end-to-end adversarial transfer learning framework with the above four components: (1) compound feature presentation learning network F, (2) feature-wise discriminator L, (3) compound-wise discriminator G, and (4) domain-wise classifier S. We solve for *TAc-fc* through optimizing the following loss function

$$\mathcal{L}(\Omega, \Theta, \Psi, \Phi) = -\lambda[(\mathcal{L}_{(l)}(\Omega, \Theta) + \mathcal{L}_{(g)}(\Omega, \Psi)] + \mathcal{L}_{(c)}(\Omega, \Phi) \tag{13}$$

where Ω, Θ, Ψ, and Φ are learnable parameters of F, L, G, and S, respectively, and λ is a trade-off parameter. This loss function combines the three loss functions for L, G, and S and will be optimized in an adversarial way as follows:

(Step 1). Minimize $\mathcal{L}$ with respect to Ω and Φ via solving the following optimization problem:

$$\hat{\Omega}, \hat{\Phi} = \underset{\Omega,\Phi}{\operatorname{argmin}}\, \mathcal{L}(\Omega, \Theta, \Psi, \Phi) \tag{14}$$

By minimizing $\mathcal{L}$, we essentially minimize $\mathcal{L}_{(c)}$ and maximize $\mathcal{L}_{(l)}$ and $\mathcal{L}_{(g)}$. As discussed in L (Section 4.2.5.1) and G (Section 4.2.5.2), maximizing $\mathcal{L}_{(l)}$ and $\mathcal{L}_{(g)}$ will encourage learning of transferable features and compounds that can be used to help the tasks; as discussed in S (Section 4.2.3), minimizing $\mathcal{L}_{(c)}$ will prevent the transfer of conflicting information, in addition to minimizing the classification errors in each task.

(Step 2). Maximize $\mathcal{L}$ with respect to Θ and Ψ via solving the following optimization problem:

$$\hat{\Theta}, \hat{\Psi} = \underset{\Theta,\Psi}{\operatorname{argmax}}\, \mathcal{L}(\Omega, \Theta, \Psi, \Phi) \tag{15}$$

By maximizing $\mathcal{L}$, we essentially minimize $\mathcal{L}_{(l)}$ and $\mathcal{L}_{(g)}$ ($\mathcal{L}_{(c)}$ is fixed in this step). As discussed in L (Section 4.2.5.1) and G (Section 4.2.5.2), minimizing $\mathcal{L}_{(l)}$ and $\mathcal{L}_{(g)}$ will encourage that L and G accurately learn features and compounds that are specific to each domain to improve the classification performance of each domain.

(Step 3). The above two steps are iterated until the learning converges.

Thus, the optimization problem consists of a maximization with respect to some parameters and a minimization with respect to the others. In order to tackle such a mini-max optimization, we insert the gradient reversal layer (GRL)[64]

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

between F and the discriminators L and G. GRL reverses the gradients during the backward propagation and hence optimizes parameters Ω by maximizing the discriminator loss.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.1c06805.

Additional details of *dmpn*, tables with information on processed assays, tables with information on assay pairs, additional experimental details with hyperparameter configurations, tables presenting similarity analysis of compounds with correct predictions by *TAc−c*, additional parameter study discussion with figures, and details on the developed compound prioritization method (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Xia Ning − *Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, United States; Biomedical Informatics and Translational Data Analytics Institute, The Ohio State University, Columbus, Ohio 43210, United States;* orcid.org/0000-0002-6842-1165; Email: ning.104@osu.edu

Authors

Vishal Dey − *Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, United States*

Raghu Machiraju − *Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, United States; Biomedical Informatics and Translational Data Analytics Institute, The Ohio State University, Columbus, Ohio 43210, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.1c06805

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This project was made possible, in part, by support from the National Science Foundation under Grant Numbers IIS-1855501, IIS-1827472, and IIS-2133650 and the AWS Machine Learning Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

■ REFERENCES

(1) Dickson, M.; Gagnon, J. P. The cost of new drug discovery and development. *Discovery Med.* **2009**, *4*, 172−179.

(2) DiMasi, J. A.; Hansen, R. W.; Grabowski, H. G. The price of innovation: New estimates of drug development costs. *J. Health Econ.* **2003**, *22*, 151−185.

(3) Terstappen, G. C.; Reggiani, A. In silico research in drug discovery. *Trends Pharmacol. Sci.* **2001**, *22*, 23−26.

(4) Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E. W. Computational methods in drug discovery. *Pharmacol. Rev.* **2014**, *66*, 334−395.

(5) Hansch, C.; Maloney, P. P.; Fujita, T.; Muir, R. M. Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. *Nature* **1962**, *194*, 178−180.

(6) Debnath, A. K.; Lopez de Compadre, R. L.; Debnath, G.; Shusterman, A. J.; Hansch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.* **1991**, *34*, 786−797.

(7) Superti-Furga, G.; Courtneidge, S. A. Structure-function relationships in Src family and related protein tyrosine kinases. *BioEssays* **1995**, *17*, 321−330.

(8) Dudek, A.; Arodz, T.; Galvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb. Chem. High Throughput Screening* **2006**, *9*, 213−228.

(9) Imrie, F.; Bradley, A. R.; Van Der Schaar, M.; Deane, C. M. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *J. Chem. Inf. Model.* **2018**, *58*, 2319−2330.

(10) Ning, X.; Rangwala, H.; Karypis, G. Multi-assay-based structure-activity relationship models: Improving structure-activity relationship models by incorporating activity information from related targets. *J. Chem. Inf. Model.* **2009**, *49*, 2444−2456.

(11) Liu, J.; Ning, X. Multi-assay-based compound prioritization via assistance utilization: a machine learning framework. *J. Chem. Inf. Model.* **2017**, *57*, 484−498.

(12) Liu, J.; Ning, X. Differential compound prioritization via bidirectional selectivity push with power. *J. Chem. Inf. Model.* **2017**, *57*, 2958−2975.

(13) Caron, P. R; Mullican, M. D; Mashal, R. D; Wilson, K. P; Su, M. S; Murcko, M. A Chemogenomic approaches to drug discovery. *Curr. Opin. Chem. Biol.* **2001**, *5*, 464−470.

(14) Kubinyi, H.; Müller, G. *Chemogenomics in Drug Discovery: A Medicinal Chemistry Perspective*; Wiley, 2005.

(15) Harris, C. J.; Stevens, A. P. Chemogenomics: structuring the drug discovery process to gene families. *Drug Discovery Today* **2006**, *11*, 880−888.

(16) Klabunde, T. Chemogenomic approaches to drug discovery: Similar receptors bind similar ligands. *Br. J. Pharmacol.* **2007**, *152*, 5−7.

(17) Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345−1359.

(18) Cai, C.; Wang, S.; Xu, Y.; Zhang, W.; Tang, K.; Ouyang, Q.; Lai, L.; Pei, J. Transfer learning for drug discovery. *J. Med. Chem.* **2020**, *63*, 8683−8694.

(19) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.

(20) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: New data content and improved web interfaces. *Nucleic Acids Res.* **2021**, *49*, 388−395.

(21) ur Rahman, A.; Choudhary, M. I.; Thomsen, W. J. *Bioassay techniques for drug development*; CRC Press, 2001.

(22) Tareq Hassan Khan, M. Predictions of the ADMET properties of candidate drug molecules utilizing different QSAR/QSPR modelling approaches. *Curr. Drug Metab.* **2010**, *11*, 285−295.

(23) Lo, Y. C.; Rensi, S. E.; Torng, W.; Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today* **2018**, *23*, 1538−1546.

(24) Katritzky, A. R.; Petrukhin, R.; Tatham, D.; Basak, S.; Benfenati, E.; Karelson, M.; Maran, U. Interpretation of quantitative structure - property and - activity relationships. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 679−685.

(25) Czerminski, R.; Yasri, A.; Hartsough, D. Use of support vector machine in pattern classification: Application to QSAR studies. *Quant. Struct.-Act. Relat.* **2001**, *20*, 227−240.

(26) Zernov, V. V.; Balakin, K. V.; Ivaschenko, A. A.; Savchuk, N. P.; Pletnev, I. V. Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 2048−2056.

(27) Hou, T.; Wang, J.; Li, Y. ADME evaluation in drug discovery. 8. The prediction of human intestinal absorption by a support vector machine. *J. Chem. Inf. Model.* **2007**, *47*, 2408−2415.
(28) Alvarsson, J.; Lampa, S.; Schaal, W.; Andersson, C.; Wikberg, J. E.; Spjuth, O. Large-scale ligand-based predictive modelling using support vector machines. *J. Cheminf.* **2016**, *8*, 1−9.
(29) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947−1958.
(30) Zhang, Q. Y.; Aires-de Sousa, J. Random forest prediction of mutagenicity from empirical physicochemical descriptors. *J. Chem. Inf. Model.* **2007**, *47*, 1−8.
(31) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of kinase inhibitors using a Bayesian model. *J. Med. Chem.* **2004**, *47*, 4463−4470.
(32) Chen, L.; Li, Y.; Zhao, Q.; Peng, H.; Hou, T. ADME evaluation in drug discovery. 10. Predictions of P-glycoprotein inhibitors using recursive partitioning and naive bayesian classification techniques. *Mol. Pharmaceutics* **2011**, *8*, 889−900.
(33) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(34) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1273−1280.
(35) Randić, M. Novel molecular descriptor for structure-property studies. *Chem. Phys. Lett.* **1993**, *211*, 478−483.
(36) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.
(37) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep learning in drug discovery. *Mol. Inf.* **2016**, *35*, 3−14.
(38) Zhang, L.; Tan, J.; Han, D.; Zhu, H. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today* **2017**, *22*, 1680−1685.
(39) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discovery Today* **2018**, *23*, 1241−1250.
(40) Hop, P.; Allgood, B.; Yu, J. Geometric deep learning autonomously learns chemical features that outperform those engineered by domain experts. *Mol. Pharmaceutics* **2018**, *15*, 4371−4377.
(41) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. DeepTox: Toxicity prediction using deep learning. *Toxicol. Lett.* **2017**, *280*, S69.
(42) Wenzel, J.; Matter, H.; Schmidt, F. Predictive multitask deep neural network models for ADME-Tox properties: Learning from Large Data Sets. *J. Chem. Inf. Model.* **2019**, *59*, 1253−1268.
(43) Wu, Z.; Lei, T.; Shen, C.; Wang, Z.; Cao, D.; Hou, T. ADMET evaluation in drug discovery. 19. Reliable prediction of human Cytochrome P450 inhibition using artificial intelligence approaches. *J. Chem. Inf. Model.* **2019**, *59*, 4587−4601.
(44) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems* **2015**, 2224−2232.
(45) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.
(46) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. *International Conference on Machine Learning* **2017**, 1263−1272.
(47) Sun, M.; Zhao, S.; Gilvary, C.; Elemento, O.; Zhou, J.; Wang, F. Graph convolutional networks for computational drug development and discovery. *Briefings Bioinf* **2020**, *21*, 919−935.
(48) Withnall, M.; Lindelöf, E.; Engkvist, O.; Chen, H. Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *J. Cheminf.* **2020**, *12*, 1−18.
(49) Zheng, S.; Yan, X.; Yang, Y.; Xu, J. Identifying structure−property relationships through SMILES syntax analysis with self-attention mechanism. *J. Chem. Inf. Model.* **2019**, *59*, 914−923.
(50) Chakravarti, S. K.; Alla, S. R. M. Descriptor free QSAR modeling using deep learning with long short-term memory neural networks. *Front. Artif. Intell.* **2019**, *2*, 17.
(51) Karpov, P.; Godin, G.; Tetko, I. V. Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *J. Cheminf.* **2020**, *12*, 1−12.
(52) Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. *A survey on deep transfer learning*; International Conference on Artificial Neural Networks, 2018; pp 270−279.
(53) Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2021**, *109*, 43−76.
(54) Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems* **2014**, 3320−3328.
(55) Wang, Z.; Dai, Z.; Poczos, B.; Carbonell, J. Characterizing and avoiding negative transfer. *IEEE Conference on Computer Vision and Pattern Recognition* **2019**, 11285−11294.
(56) Pan, S. J.; Tsang, I. W.; Kwok, J. T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2011**, *22*, 199−210.
(57) Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P. S. Transfer feature learning with joint distribution adaptation. *IEEE International Conference on Computer Vision* **2013**, 2200−2207.
(58) Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **2018**, *312*, 135−153.
(59) Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *CoRR* **2014**, abs/1412.3474.
(60) Long, M.; Cao, Y.; Wang, J.; Jordan, M. I. Learning transferable features with deep adaptation networks. *International Conference on Machine Learning*; Lille: France, 2015; pp 97−105.
(61) Long, M.; Zhu, H.; Wang, J.; Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. *Advances in Neural Information Processing Systems* **2016**, 136−144.
(62) Liu, H.; Long, M.; Wang, J.; Jordan, M. I. Transferable adversarial training: A general approach to adapting deep classifiers. *International Conference on Machine Learning* **2019**, 4013−4022.
(63) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems* **2014**, 2672−2680.
(64) Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. *International Conference on Machine Learning*; Lille: France, 2015; pp 1180−1189.
(65) Long, M.; Zhu, H.; Wang, J.; Jordan, M. I. *Deep transfer learning with joint adaptation networks*; International Conference on Machine Learning, 2017; pp 2208−2217.
(66) Sankaranarayanan, S.; Balaji, Y.; Castillo, C. D.; Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition*; 2018; pp 8503−8512.
(67) Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discriminative domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
(68) Long, M.; Cao, Z.; Wang, J.; Jordan, M. I. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems* **2018**, 1640−1650.
(69) Monteiro, J.; Gibert, X.; Feng, J.; Dumoulin, V.; Lee, D.-S. Domain conditional predictors for domain adaptation. *Proc. Mach. Learn. Res.* **2021**, *148*, 193−220.
(70) Pei, Z.; Cao, Z.; Long, M.; Wang, J. Multi-adversarial domain adaptation. *AAAI Conference on Artificial Intelligence*, 2018.
(71) Zhao, H.; Zhang, S.; Wu, G.; Costeira, J. P.; Moura, J. M.; Gordon, G. J. Multiple source domain adaptation with adversarial

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483

learning. *International Conference on Learning Representations - Workshop Track Proceedings*, 2018.
(72) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low data drug discovery with one-shot learning. *ACS Cent. Sci.* **2017**, *3*, 283−293.
(73) Simões, R. S.; Maltarollo, V. G.; Oliveira, P. R.; Honorio, K. M. Transfer and multi-task learning in QSAR modeling: Advances and challenges. *Front. Pharmacol.* **2018**, DOI: 10.3389/fphar.2018.00074.
(74) Lee, M.; Kim, H.; Joe, H.; Kim, H. G. Multi-channel PINN: Investigating scalable and transferable neural networks for drug discovery. *J. Cheminf.* **2019**, *11*, 1−16.
(75) Guo, Z.; Zhang, C.; Yu, W.; Herr, J.; Wiest, O.; Jiang, M.; Chawla, N. V. *Few-shot graph learning for molecular property prediction*; Web Conference 2021: New York, NY, USA, 2021; pp 2559−2567.
(76) Dahl, G. E.; Jaitly, N.; Salakhutdinov, R. Multi-task neural networks for QSAR predictions. *CoRR* **2014**, abs/1406.1231.
(77) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is multitask deep learning practical for pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.
(78) Xu, Y.; Ma, J.; Liaw, A.; Sheridan, R. P.; Svetnik, V. Demystifying multitask deep neural networks for quantitative structure−activity relationships. *J. Chem. Inf. Model.* **2017**, *57*, 2490−2504.
(79) Rodríguez-Pérez, R.; Bajorath, J. Multitask machine learning for classifying highly and weakly potent kinase inhibitors. *ACS Omega* **2019**, *4*, 4367−4375.
(80) Sosnin, S.; Vashurina, M.; Withnall, M.; Karpov, P.; Fedorov, M.; Tetko, I. V. A survey of multi-task learning methods in chemoinformatics. *Mol. Inf.* **2019**, *38*, 1800108.
(81) Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. *Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction*; ACM SIGKDD International Conference on Knowledge Discovery & Data Mining: New York, NY, USA, 2018; pp 302−310.
(82) Li, X.; Fourches, D. Inductive transfer learning for molecular activity prediction: Next-gen QSAR models with MolPMoFiT. *J. Cheminf.* **2020**, *12*, 1−15.
(83) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput* **1997**, *9*, 1735−1780.
(84) Abbasi, K.; Poso, A.; Ghasemi, J.; Amanlou, M.; Masoudi-Nejad, A. Deep transferable compound representation across domains and tasks for low data drug discovery. *J. Chem. Inf. Model.* **2019**, *59*, 4528−4539.
(85) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: Improved access to chemical data. *Nucleic Acids Res.* **2019**, *47*, 1102−1109.
(86) Bateman, A.; et al. UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Res.* **2021**, *49*, 480−489.
(87) Santos, R.; Ursu, O.; Gaulton, A.; Bento, A. P.; Donadi, R. S.; Bologa, C. G.; Karlsson, A.; Al-Lazikani, B.; Hersey, A.; Oprea, T. I.; Overington, J. P. A comprehensive map of molecular drug targets. *Nat. Rev. Drug Discovery* **2017**, *16*, 19−34.
(88) Zdrazil, B.; Richter, L.; Brown, N.; Guha, R. Moving targets in drug discovery. *Sci. Rep.* **2020**, *10*, 1−15.
(89) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical similarity searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(90) Landrum, G. *RDKit: open-source cheminformatics software*. 2020; https://www.rdkit.org (accessed 2020-01-22).
(91) Wang, X.; Gao, J.; Long, M.; Wang, J. *Self-tuning for data-efficient deep learning*; International Conference on Machine Learning, 2021; pp 10738−10748.
(92) Kingma, D. P.; Ba, J. L. Adam: A method for stochastic optimization. *International Conference on Learning Representations*; 2015.
(93) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.
(94) *Descriptor computation (chemistry) and (optional) storage for machine learning*. https://github.com/bp-kelley/descriptastorus (accessed 2020-08-10).
(95) Scarselli, F.; Gori, M.; Ah Chung Tsoi; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61−80.
(96) Veličković, P.; Casanova, A.; Liò, P.; Cucurull, G.; Romero, A.; Bengio, Y. Graph attention networks. *CoRR* **2018**, abs/1710.10903.

https://doi.org/10.1021/acsomega.1c06805
ACS Omega 2022, 7, 9465−9483