# Adaptive Invariant Risk Minimization for Molecule Property Prediction

**Wengong Jin**  **Regina Barzilay**  **Tommi Jaakkola**
CSAIL, Masssachusetts Institute of Technology
{wengong,regina,tommi}@csail.mit.edu

## Abstract

## 1 Introduction

The current race to identify promising repurposing drug candidates for COVID-19 highlights a significant limitation of existing property prediction algorithms. Since their accuracy tightly depends on the access to large amounts of in-domain training data, these algorithms provide limited utility today when high-throughput screening is lacking. At the time of writing, there are only 48 drugs and 880 fragments with measured in-vitro SARS-CoV-2 activity in the public domain [6]. This scenario is not unique to the current pandemic, but is also common for initial stages of novel therapeutic development where limited screening data has to inform on-going experimental exploration.

Ideally, we need to supplement this scarce data with other related data sources. In the COVID-19 case, one source of such data is SARS-CoV-1 screens. Since SARS-CoV-1 and SARS-CoV-2 proteases are similar (79.5% sequence identity) [11], drugs screened against SARS-CoV-1 is expected to be relevant for SARS-CoV-2 predictions. The main challenge, however, is the distributional mismatch between two data sources. The 880 fragments screened against SARS-CoV-2 protease consist of only 14.2 atoms on average, constituting 37.2% of full drug size molecules. To effectively utilize informative but distributionally different data, we need to develop an algorithm that can robustly generalize across different domains.

One natural framework for reasoning about extrapolation is invariant risk minimization (IRM) [1], which aims to find a predictor that is simultaneously optimal across different domains. The IRM introduces the notion of environment to explicate "nuisance" variances which predictor should ignore. In molecular space, the notion of environment is linked to scaffolds [2] which succinctly represent chemical space. However, many environments may have only one molecule due to combinatorial nature of scaffolds. This makes the IRM independence criteria vacuous as the environment uniquely specifies the input.

To address this challenge, we propose a novel method which constructs two environments for IRM. The constructed environments share the same set of training examples, but differ in terms of the representation the predictor operates on. Specifically, we introduce a perturbation $\delta(x)$ over molecular representation $\phi(x)$ constructed with the help of the scaffold classifier. The perturbation is applied in one environment while the other one remains the same. In this way, we aim to explicate directions in the latent representation that the predictor should be invariant to. Importantly, our approach does not divide examples into disjoint environments and works with sparse environments with high cardinality.

Our method is evaluated on existing SARS-CoV-2 screening data [6]. We compare against multiple transfer learning techniques such as domain adversarial training [5] and conditional domain adversarial network [7]. On two SARS-CoV-2 datasets, the proposed approach outperforms the best performing baseline with 8-16% relative AUROC improvement. Finally, we apply our model on Broad drug repurposing hub [4] and report the top 20 predictions for further investigation.

## 2 Domain Extrapolation

Training data in many emerging applications is necessarily limited, fragmented, or otherwise heterogeneous. It is therefore important to ensure that model predictions derived from such data generalize substantially beyond where the training samples lie. In other words, the trained model should have the ability to extrapolate. For instance, in computational chemistry, it is desirable for property prediction models to perform well in time-split scenarios where the evaluation concerns compounds that were created after those in the training set. Another way to simulate evaluation on future compounds is through a scaffold split [2]. A scaffold split between training and test introduces some structural separation between the chemical spaces of the two sets of compounds, hence evaluating the model's ability to extrapolate to a new domain.

One way to ensure domain extrapolation is to enforce an appropriate invariance criterion during training. We envision here that the compounds $X$ can be divided into potentially a large number of domains or "environments" $E$, for example, based on their scaffold. The goal is then to learn a parametric mapping $Z = \phi(X)$ of compounds $X$ to their latent representations $Z$ in a manner that satisfies the chosen invariance criterion. A number of such strategies relevant to extrapolation have been proposed. They can be roughly divided into the following three categories:

- Domain adversarial training [5] enforces the latent representation $Z = \phi(X)$ to have the same distribution across different domains $E$. If we denote by $P(X|E)$ the conditional distribution of compounds in environment $E$, then we want $P(\phi(X)|E) = P(\phi(X))$ for all $E$. With some abuse of notation, we can write this condition as $Z \perp E$. A single predictor is learned based on $Z = \phi(X)$, i.e., all the domains share the same predictor. As a result, the predicted label distribution $P(Y)$ will also be the same across the domains. This can be problematic when the training and test domains have very different label distributions [10]. The independence condition itself can be challenging to satisfy when the chemical spaces overlap across the environments.
- Conditional domain adaptation [7] relaxes the requirement that the label distributions must agree across the environments. The key idea is to condition the invariance criterion on the label. In other words, we require that $P(\phi(X)|E, Y) = P(\phi(X)|Y)$ for all $E$ and $Y$, i.e., we aim to satisfy the independence statement $Z \perp E \mid Y$. The formulation allows the label distribution to vary between domains since $E$ and $Y$ can depend on each other. The constraint remains, however, too restrictive about the latent representation. To illustrate this, consider a simple case where the environments share the same chemical space and differ only in terms of proportions of different types of compounds in them. These type proportions play roles analogous to label proportions in domain adversarial training. Hence, the only way to achieve $Z \perp E \mid Y$ would be if the proportions were the same across environments. To state the example differently, a functional mapping $Z = \phi(X)$ cannot fractionally assign probability mass placed on $X$ to different latent space locations $Z$; it all has to be mapped to a single location. To reduce the impact of the strict condition, we would have to introduce $P(Z|\phi(X), E)$ in place of the simpler functional mapping $Z = \phi(X)$, further complicating the approach.
- Invariant risk minimization (IRM) [1] seeks a different notion of invariance, focusing less on aligning distributions of latent representations, and instead shifting the emphasis on how those representations can be consistently used for predictions. The IRM principle requires that the predictor $f$ operating on $Z = \phi(X)$ is simultaneously optimal across different environments or domains. For example, this holds if our representation explicates only features that are (causally) necessary for the correct prediction. How $\phi(X)$ is distributed across the environments is then immaterial. The associated conditional independence criterion is $Y \perp E \mid Z$. In other words, knowing the environment shouldn't provide any additional information about $Y$ beyond the features $Z = \phi(X)$. The distribution of labels can differ across the environments.

While the IRM principle provides a natural framework for domain extrapolation, it needs to be extended in several ways for our setting. The main limitation of the original framework is that the environments $E$ themselves are fixed and pre-defined. Their role in the pricinple is to illustrate "nuisance" variation, i.e., variability that the predictor should learn not to rely on. In order to enforce the associated independence criterion, we need a fair number of examples within each such environment. The approach therefore becomes unsuitable when the natural environments such as scaffolds are combinatorially defined or otherwise have high cardinality. Indeed, we might have only a single molecule per environment in our training set, making the independence criterion vacuous ($E$ would uniquely specify $X$, thus also $Z$ and $Y$). A straightforward remedy for the high cardinality

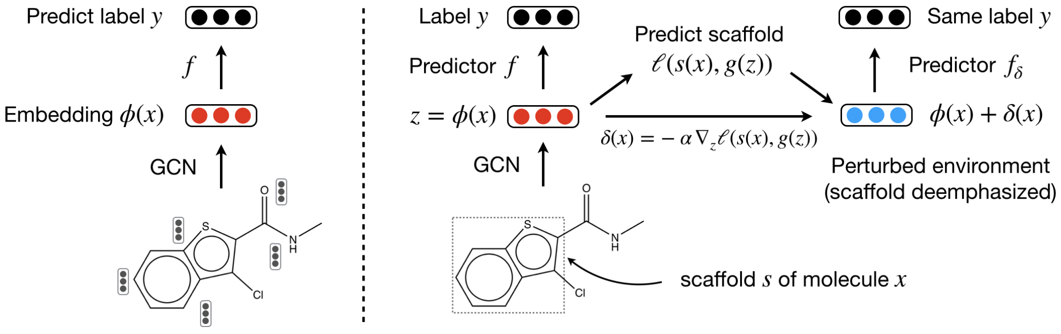

Figure 1: **Left**: Illustration of base GCN model for molecule property prediction. **Right**: IRM with adaptive environments. The model perturbs the representation $\phi(x)$ into $\phi(x) + \delta(x)$ using the gradient from scaffold classifier $g$. The predictor $f$ is trained to be simultaneously optimal across two environments. The scaffold $s$ is a subgraph of a molecule $x$ with its side chains removed [2].

environments would be to introduce a coarser definition, and enforce the principle at this coarse level instead. Since environments represent constraints on the predictor, their role in estimation is adversarial. What is then the appropriate trade-off between such a coarser definition (relaxation of constraints) and our ability to predict? We side-step having to answer this question, and instead propose to dynamically map the large number of environments to just two. These two environments are designed to nevertheless highlight the nuisance variation the predictor should avoid but do so in a tractable manner.

## 3 IRM with adaptive environments

Our goal is to adaptively highlight to the predictor the type of variability that it ought not to rely on. We do this by replacing high cardinality environments such as those based on scaffold with just two new environments. These two new environments are unusual in the sense that they share the exact same set of examples. Indeed, they only differ in terms of the representation that the predictor operates on. The first environment simply corresponds to the representation we are trying to learn, i.e., $z = \phi(x)$, where the lowercase letters refer to specific instances rather than random variables. The second environment is defined in terms of a modified representation $\phi(x) + \delta(x)$ that is a perturbed version of $\phi(x)$ and constructed with the help of the environment or scaffold classifier. The goal of $\delta(x)$ is to explicate directions in the latent representation that the predictor should avoid paying attention to. While traditional IRM environments divide examples $x$ into environments, often exclusively, we instead exercise different latent representations over the same set of examples.

More formally, our two environments correspond to a choice of perturbation $h \in \{0, \delta\}$ used to derive the latent representation $z$ from $x$, i.e., $\phi(x) + h(x)$. The associated target labels are clearly the same regardless of which perturbation (none or $\delta(x)$) was chosen. The key part of our approach pertains to how $\delta(x)$ is defined. To this end, let $g(\phi(x)))$ be a parametric environment classifier that we will instantiate in detail later. The associated classification loss is $\ell(s(x), g(\phi(x)))$ where $s(x)$ is the correct original environment label (here a scaffold) for $x$. The scaffold classifier is evolved together with the feature mapping $\phi$ and the associated predictor $f$. We define the non-zero perturbation $\delta(x)$ in terms of the negative gradient:

$$\delta(x) = -\alpha\nabla_z\ell(s(x), g(z))_{|z=\phi(x)} \tag{1}$$

where $\alpha$ is a step size parameter. The goal of this perturbation is to turn $\phi(x)$ into its "generic" version $\phi(x) + \delta(x)$ by adjusting against the environment classifier. Note that if we were to perform adversarial domain alignment, $\delta(x)$ would represent a reverse gradient update to modify $\phi(x)$. We do not do that, instead we are using the perturbation to highlight directions of variability to avoid for the predictor $f$ within an overall IRM formulation. The degree to which $\phi(x)$ is adjusted in response to $\delta(x)$ arises from the IRM principle, not from a direct alignment objective.

We begin by building the overall training objective which is then optimized in batches as described in Algorithm 1. Let $(x_i, y_i)$ be a pair of training example + the associated label to predict. Each $x_i$

**Algorithm 1** Training IRM with adaptative environments

```
for each training step do
    Sample a batch of molecules $\{x_1, \cdots, x_n\}$ with their environments $\{s_1, \cdots, s_n\}$
    Encode molecules $\{x_1, \cdots, x_n\}$ into vectors $\{\phi(x_1), \cdots, \phi(x_n)\}$
    Computed environment classification loss $\mathcal{L}^s(g \circ \phi) = \sum_i \ell(s_i, g(\phi(x_i)))$.
    Construct perturbed representation $\delta(x_i) = -\alpha \nabla_{\phi(x_i)} \ell(s_i, g(\phi(x_i)))$
    Compute invariant predictor loss $\mathcal{L}(f \circ \phi)$
    Compute competing predictor loss $\mathcal{L}(f_h \circ (\phi + h)) \quad h \in \{0, \delta\}$
    Update $\phi, f, g$ to minimize Eq. (5)
    Update $f_0, f_\delta$ to maximize Eq. (5)
end for
```

also has an environment label/features given by $s_i = s(x_i)$ (the original mapping of examples to environments is assumed given and fixed, defined by $s(x)$). The environment classifier is trained to minimize

$$\mathcal{L}^e(g \circ \phi) = \sum_i \ell(s_i, g(\phi(x_i))) \tag{2}$$

As we will explain later on, the environment classifier remains "unaware" of how the perturbation is derived on the basis of its predictions. The loss of the predictor $f$, now operating on $\phi(x) + h(x)$, where $h \in \{0, \delta\}$, is defined as

$$\mathcal{L}(f \circ (\phi + h)) = \sum_i \ell\big(y_i, f(\phi(x_i) + h(x_i))\big) \qquad h \in \{0, \delta\} \tag{3}$$

The specific form of the loss depends on the prediction task. In accordance with the IRM principle, we enforce that the predictor operating on $z = \phi(x) + h(x)$ remains optimal whether its input is $\phi(x)$ or the perturbed version $\phi(x) + \delta(x)$. In other words, we require that

$$\mathcal{L}(f \circ (\phi + h)) \leq \min_{h \in \{0, \delta\}} \mathcal{L}(f_h \circ (\phi + h)) \tag{4}$$

where $f_h$ is a predictor in the same parametric family as $f$ but trained separately with the knowledge of $h$ (perturbed or not). By relaxing the constraints via Lagrange multipliers, we express the overall training objective as

$$\lambda_e \mathcal{L}^e(g \circ \phi) + \sum_{h \in \{0, \delta\}} \Big[ (1 + \lambda_h) \mathcal{L}(f \circ (\phi + h)) - \lambda_h \mathcal{L}(f_h \circ (\phi + h)) \Big] \tag{5}$$

This minimax objective is *minimized* with respect to $\phi$, $g$, and $f$, and *maximized* with respect to $f_h$, $h \in \{0, \delta\}$. A few remarks are necessary concerning this objective:

- Even though $\delta$ is defined on the basis of $\phi$ and the environment classifier $g$, we view it as a functionally independent player. The goal of $\delta$ is to enforce optimality of $f$ and therefore it plays an adversarial role relative to $f$. Similarly to GAN objectives where the discriminator has a separate objective function, different from the generator, we separate out $\delta$ as another player in an overall game theoretic objective. Specifically, $\delta$ takes input from $\phi$ and $g$ but does not inform them in return in back-propagation.[1]
- $\phi$ in our objective is adjusted to also help the auxiliary environment classifier. This is contrary to domain alignment where the goal would be to take out any dependence on the environment. The benefit in our formulation is two-fold. First, the term grounds $\phi$ also based on the auxiliary objective, helping it to retain useful information about each example $x$. Second, the term grounds and stabilizes the definition of $\delta$ as the gradient of the environment predictor since $g$ no longer approaches a random predictor. It would be weak if $\phi$ contains no information about the environment as in domain alignment. Thus $\delta$ remains well-defined as a direction throughout the optimization.

The training procedure is shown in Algorithm 1.

[1] Incorporating this higher order dependence would not improve the empirical results.

### 3.1 Adapting the framework to molecule property prediction

In molecule property prediction, the training data is a collection of pairs $\{(x_i, y_i)\}$, where $x_i$ is a molecular graph and $y_i$ is its activity score, typically binary (active/inactive). The feature extractor $\phi(\cdot)$ is a graph convolutional network (GCN) which translates a molecular graph into a continuous vector through directed message passing operations [9]. The predictor $f$ is a feed-forward network that takes $\phi(x)$ or $\phi(x) + \delta(x)$ as input and yields predicted activity $y$.

The original environment of each compound $x_i$ is defined as its Murcko scaffold [2], which is a subgraph of $x_i$. Since scaffold is a combinatorial object with a large vocabulary of possible values, we define and train the environment classifier in a contrastive fashion [8]. Specifically, for a given molecule $x_i$ with scaffold $s_i$, we randomly sample $n$ other molecules and take their associated scaffolds $\{s_k\}$ as negative examples, as the contrastive set $C$. The environment classifier $g$ makes use of a feed-forward network $g_s$ that maps each compound or a scaffold (subgraph) to a feature vector. The probability that $x_i$ is mapped to its correct scaffold $s_i$ is then defined as

$$P_g(s = s_i | x_i, C) = \frac{\exp\{\mathrm{sim}\big(g_s(\phi(x_i)), g_s(\phi(s_i))\big)\}}{\sum_{k\in\{i\}\cup C} \exp\{\mathrm{sim}\big(g_s(\phi(x_i)), g_s(\phi(s_k))\big)\}} \tag{6}$$

where $\mathrm{sim}(\cdot, \cdot)$ stands for cosine similarity. In practice, we use the molecules within the same batch as negative examples.

## 4 Experiments

Our experiments consist of two setting. To compare our method with existing transfer learning techniques, we first evaluate our methods on a standard unsupervised transfer setup. All the models are trained on SARS-CoV-1 data and tested on SARS-CoV-2 compounds. Next, in order to select drug candidates for SARS-CoV-2, we extend our method by incorporating labeled SARS-CoV-2 data to maximize prediction accuracy and perform virtual screening over Broad drug repurposing hub [4].

**Training data** Our training data consist of three screens related to SARS-CoV. All the data can be found at `https://github.com/yangkevin2/coronavirus_data`.

- **SARS-CoV-2 MPro inhibition** 881 fragments screened for SARS-CoV-2 main protease (Mpro) collected by the Diamond Light Source group[2]. The dataset contains 78 hits.
- **SARS-CoV-2 antiviral activity** 48 FDA-approved drugs screened for antiviral activity against SARS-CoV-2 in vitro [6], including reference drugs such as Remdesivir, Lopinavir and Chloroquine. The dataset contains 27 hits.
- **SARS-CoV-1 3CLpro inhibition** Over 290K molecules screened for activity against SARS-CoV-1 3C-like protease (3CLpro) in PubChem AID1706 assay. There are 405 active compounds.

**Baselines** We compare the proposed approach with the following baselines:

- **Direct transfer**: We train a GCN on SARS-CoV-1 data and directly test it on SARS-CoV-2 data.
- **Domain adversarial training** (DANN) [5]: Since distribution of molecules is different between SARS-CoV-1 and SARS-CoV-2 datasets, we use domain adversarial training to facilitate transfer. Specifically, we augment our GCN with additional domain classifier $g$ to enforce the distribution of $\phi(x)$ to be the same across training (SARS-CoV-1) and test set (SARS-CoV-2).
- **Conditional adversarial domain adaptation** (CDAN) [7] conditions the domain classifier $g$ with predicted labels $f(\phi(x))$. In particular, we adopt their multilinear conditioning strategy: the input to $g$ becomes a vector outer-product $\phi(x) \otimes f(\phi(x))$, which has the same dimension as $\phi(x)$ for binary classification tasks.
- **Scaffold adversarial training** (SANN): This is an extension of DANN where the domain classifier $g$ is replaced with our scaffold classifier $g_s$ in Eq.(6). SANN seeks to learn a scaffold-invariant representation $\phi(x)$ through the following minimax game ($\mathcal{L}^e$ is scaffold classification loss):

$$\min_{\phi,f} \max_{g} \mathcal{L}(f \circ \phi) - \lambda_e \mathcal{L}^e(g_s \circ \phi) \tag{7}$$

[2] `www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem`

Table 1: Unsupervised transfer results. Models are trained on SARS-CoV-1 data and tested on SARS-CoV-2 compounds. We report average of AUROC under 5-fold cross validation.

| | Mpro inhibition AUC | Antiviral activity AUC |
|---|---|---|
| Direct Transfer | $0.642 \pm 0.021$ | $0.415 \pm 0.081$ |
| DANN [5] | $0.646 \pm 0.012$ | $0.607 \pm 0.121$ |
| SANN | $0.630 \pm 0.079$ | $0.570 \pm 0.096$ |
| CDAN [7] | $0.625 \pm 0.013$ | $0.639 \pm 0.067$ |
| IRM [1] | $0.653 \pm 0.022$ | $0.391 \pm 0.086$ |
| Our method | $\mathbf{0.756 \pm 0.012}$ | $\mathbf{0.695 \pm 0.068}$ |
| - without IRM | $0.736 \pm 0.007$ | $0.678 \pm 0.057$ |

- **Invariant risk minimization** (IRM): The original IRM [1] requires the predictor $f$ to be constant, which does not work well in our setting. Therefore, we adopt an adversarial formulation for IRM proposed in [3], allowing us to use powerful neural predictors:

$$\min_{\phi,f} \max_{f_1,\cdots,f_K} \sum_e (1+\lambda_e)\mathcal{L}(f \circ \phi) - \lambda_e \mathcal{L}(f_e \circ \phi) \tag{8}$$

  Here each of the $K$ environments consists of molecules with the same scaffold. Since the number of environments is large, we impose parameter sharing among the competing predictors $f_1, \cdots, f_K$. Specifically, the input of $f_j = f'(\phi(x), j)$ is a concatenation of $\phi(x)$ and one-hot encoding of $j$.

**Model hyperparameters** For our model, we set $\lambda_h = \lambda_e = 0.1$ and perturbation learning rate $\alpha = 0.1$, which worked well across all experiments. All methods are trained with Adam using its default configuration. Our GCN implementation is based on chemprop [9].

- For unsupervised transfer, we use their default hyper-parameter setting. For all methods, the GCN $\phi$ has three layers with hidden layer dimension 300. The predictor $f$ is a two-layer MLP.
- For supervised transfer, we perform hyper-parameter optimization to identify the best architecture for the multitask GCN. The GCN $\phi$ has two layers with hidden layer dimension 2000. The predictor $f$ is a three-layer MLP. The dropout rate is 0.1. For fair comparison, all the methods use the same architecture in this setting.

### 4.1 Unsupervised transfer

**Setup** Our model is a single-task binary classification model which predicts the SARS-CoV-1 3CLpro inhibition. After training, the model is tested on Mpro and antiviral dataset. Each model is evaluated under five independent runs and we report the average AUROC score on the test set.

**Results** Our results are shown in Table 1. The proposed method significantly outperformed all the baselines, especially on the Mpro inhibition prediction dataset (0.756 versus 0.653 AUROC).

**Ablation study** Indeed, the improvement of our model comes from two sources: the additional auxiliary task and IRM principle. To show individual contribution of each component, we conduct an ablation study of our method without the IRM principle. The loss function in this case is the scaffold classification loss plus property prediction loss $\lambda_e \mathcal{L}^e(g \circ \phi) + \mathcal{L}(f \circ \phi)$. The performance of this method is shown in the end of Table 1 ("without IRM"). The auxiliary scaffold classifier shows quite significant improvement, but is still inferior to our full model trained with IRM principle.

### 4.2 Drug repurposing for SARS-CoV-2

**Setup** We extend all the methods to multitask binary classification models that predict *three* different properties for each new compound: 1) probability of inhibiting the SARS-CoV-2 Mpro; 2) antiviral activity against SARS-CoV-2; 3) probability of inhibiting SARS-CoV-1 3CLpro.

Each model is evaluated under 5-fold cross validation with the same splits. In each fold, the training set contains the SARS-CoV data and 60% of the SARS-CoV-2 data (Mpro + antiviral), and the test

Table 2: 5-fold cross validation of different methods. CoV-2 means that the model is trained on SARS-CoV-2 data. CoV-1 means that the model is additionally trained on SARS-CoV-1 data.

| Method | CoV-1 | CoV-2 | Mpro inhibition AUC | Antiviral activity AUC |
|---|---|---|---|---|
| Multitask GCN | | ✓ | $0.7646 \pm 0.0398$ | $0.7404 \pm 0.1117$ |
| Multitask GCN | ✓ | ✓ | $0.7841 \pm 0.0416$ | $0.8067 \pm 0.0690$ |
| DANN | ✓ | ✓ | $0.7785 \pm 0.0321$ | $0.8146 \pm 0.1068$ |
| IRM [1] | ✓ | ✓ | $0.7778 \pm 0.0426$ | $0.8198 \pm 0.1072$ |
| Our method | ✓ | ✓ | $\mathbf{0.7955 \pm 0.0489}$ | $\mathbf{0.8930 \pm 0.0267}$ |

Table 3: Top 20 molecules in the Broad drug repurposing hub predicted by our model.

| SMILES | MPro | Antiviral |
|---|---|---|
| NS(=O)(=O)c1ccc(cc1)S(=O)(=O)Nc1cccc2c(Cl)c[nH]c12 | 0.686 | 0.504 |
| Cc1cc2CCCS(=O)(=O)c2cc1S(N)(=O)=O | 0.518 | 0.557 |
| Nc1ccc(cc1)S(=O)(=O)Nc1ccccn1 | 0.579 | 0.496 |
| Nc1ccc(cc1)S(=O)(=O)Nc1nccs1 | 0.545 | 0.506 |
| NS(=O)(=O)c1cc(Cl)c(Cl)c(c1)S(N)(=O)=O | 0.557 | 0.492 |
| Nc1ccc(cc1)S(=O)(=O)Nc1ncccn1 | 0.555 | 0.480 |
| Nc1ccc(cc1)S(=O)(=O)c1ccc(N)cc1 | 0.477 | 0.548 |
| Nc1ccc(cc1)S(=O)(=O)Nc1cnc2ccccc2n1 | 0.523 | 0.486 |
| Nc1ccc(cc1)S(N)(=O)=O | 0.652 | 0.378 |
| CC(=O)Nc1ccc(cc1)S(=O)(=O)c1ccc(NC(C)=O)cc1 | 0.473 | 0.510 |
| Nc1ccccc1S(N)(=O)=O | 0.603 | 0.398 |
| NS(=O)(=O)c1cc2c(cc1Cl)N=C(CSCc1ccccc1)NS2(=O)=O | 0.594 | 0.394 |
| CCN[C@@H]1C[C@H](C)S(=O)(=O)c2sc(cc12)S(N)(=O)=O | 0.372 | 0.629 |
| NS(=O)(=O)c1cc2c(NC(NS2(=O)=O)C(Cl)Cl)cc1Cl | 0.358 | 0.622 |
| NS(=O)(=O)NCc1csc2ccccc12 | 0.654 | 0.338 |
| CN(C(=O)C(Cl)Cl)c1ccc(OC(=O)c2ccco2)cc1 | 0.348 | 0.632 |
| Nc1ccc(cc1)S(=O)(=O)Nc1ccc(Cl)nn1 | 0.376 | 0.567 |
| CC(=O)Nc1ccc(cc1)S(=O)(=O)Nc1ccc(cc1)[N+]([O-])=O | 0.375 | 0.528 |
| COc1nccnc1NS(=O)(=O)c1ccc(N)cc1 | 0.337 | 0.585 |
| COc1cnc(NS(=O)(=O)c2ccc(N)cc2)nc1 | 0.345 | 0.552 |

set contains the rest 40% of the SARS-CoV-2 compounds. We report the mean and standard deviation of AUROC score evaluated on five different folds.

**Results** Our results are shown in Table 2. The proposed method significantly outperformed the two baselines, especially on the antiviral activity prediction dataset (0.89 versus 0.82 AUROC). As an ablation study, we also trained a GCN on only SARS-CoV-2 data (the first row in Table 2). Indeed, the multitask GCN trained with additional SARS-CoV-1 data performs better (0.740 vs 0.807 on antiviral prediction), indicating that the two virus are closely related.

The best model is then used to predict the SARS-CoV-2 Mpro inhibition and antiviral activity of compounds in Broad drug repurposing hub. In order to utilize maximal amount of labeled data, the model is re-trained under 10-fold cross validation with 90%/10% split (instead of 60%/40%). The resulting 10 models are combined together as an ensemble to predict properties for new compounds.

We report the top 20 molecules with the highest MPro inhibition plus antiviral activity score (see Table 3). To understand why these compounds are considered as positives, we further plot the nearest fragment in the Mpro dataset for each selected compound. As shown in Figure 2, we can see that many of these compounds have similar fragments in the Mpro dataset.

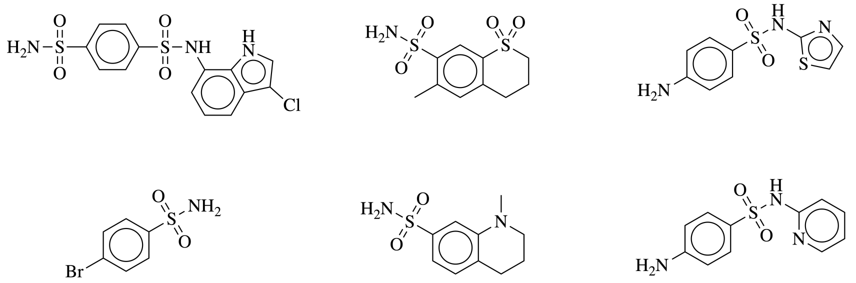

Figure 2: Selected compounds (top) and their nearest neighbor in the Mpro fragment dataset (bottom). Many of the compounds contains similar fragments in the Mpro dataset.

## 5 Conclusion

In this paper, we investigate existing domain extrapolation paradigms and their limitations. To allow the method to extraploate across combinatorially many environments, we propose a new method which complements invariant risk minimization with adaptive environments. The method is evaluated on molecule property prediction tasks and shows significant improvements over strong baselines.

## References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[3] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S Jaakkola. Invariant rationalization. *arXiv preprint arXiv:2003.09772*, 2020.

[4] Steven M Corsello, Joshua A Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi E Hirschman, Stephen E Johnston, Anita Vrcic, Bang Wong, Mariya Khan, et al. The drug repurposing hub: a next-generation drug library and information resource. *Nature medicine*, 23 (4):405–408, 2017.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[6] Sangeun Jeon, Meehyun Ko, Jihye Lee, Inhee Choi, Soo Young Byun, Soonju Park, David Shum, and Seungtaek Kim. Identification of antiviral drug candidates against sars-cov-2 from fda-approved drugs. *bioRxiv*, 2020.

[7] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[9] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59 (8):3370–3388, 2019.

[10] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

[11] Peng Zhou, Xing-Lou Yang, Xian-Guang Wang, Ben Hu, Lei Zhang, Wei Zhang, Hao-Rui Si, Yan Zhu, Bei Li, Chao-Lin Huang, et al. A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature*, 579(7798):270–273, 2020.