

---

## A NOTATIONS

Our notation is based on index notation and Einstein summation conventions. Notation of functions and matrices in our algorithm is as follows.

$$\begin{aligned}
X &: \text{Vector} \\
X^\mu &: \text{Vector Field} \\
dx_\mu &: \text{Basis} \\
X_\mu &: \text{Dual Vector Field} \\
dx^\mu &: \text{Dual Basis} \\
T &: \text{Tensor} \\
T^{\nu_1 \dots \nu_p}_{\mu_1 \dots \mu_q} &: (p, q) \text{ Tensor Field} \\
g_{\mu\nu} &: \text{Metric Tensor} \\
\delta_{\mu\nu} &: \text{Kronecker Delta} \\
\nabla_\mu &: \text{Covariant Derivative} \\
\mathcal{L}_X &: \text{Lie Derivative} \\
\Gamma^\rho_{\mu\nu} &: \text{Christoffel Symbol}
\end{aligned}$$

All indices are raised and lowered by a metric  $g_{\mu\nu}$ . For instances,

$$g^\mu{}_\nu = g^{\mu\rho} g_{\rho\nu} \quad (1)$$

where

$$g^{\mu\nu} g_{\mu\nu} = \delta^\mu{}_\nu = D \quad (2)$$

Here  $D$  is the number of dimensions.

## B PROOFS AND DERIVATIONS

### B.1 THE DEFINITION OF RIEMANNIAN MANIFOLD

A curved space is complicated to comprehend in general. Since late 19th century, there has been immense development in differential geometry to interpret curved spaces formally. One of the best-known intuitive geometries is the Riemannian. Riemannian geometry enjoys a handful of useful mathematical characters that can be utilized in the real world. The formal definition of Riemannian is as follows:

**Definition B.1** (Riemannian Manifold). A Riemannian metric on a smooth manifold  $M$  is a choice at each point  $x \in M$  of a positive definite inner product  $g_p : T_p M \times T_p M \rightarrow \mathbb{R}$  on  $T_x M$ . The smooth manifold endowed with the metric  $g$  is a Riemannian manifold, denoted  $(M, g)$ .

As it is expressed above, a Riemannian manifold is smooth and differentiable everywhere on the manifold and its derivative as well. Also, a Riemannian enjoys diffeomorphism invariances, induced by the Lie derivative  $\mathcal{L}_X$ . One can easily notice that the adjoint operation between two different Lie derivatives forms a group, namely the diffeomorphism group. This isometry ensures coordinate choices without changing the global geometry of the space.

$$X' = X'^\mu dX'_\mu = X'^\mu \frac{\partial X^\nu}{\partial X'^\mu} dX_\nu = X^\nu dX_\nu = X \quad (3)$$

As it is depicted in eq. 3, transformed vector remains unchanged. Moreover, one can always fix the transformed coordinate in a locally flat space.

$$\xi^\mu = \frac{\partial \xi^\mu}{\partial X^\nu} X^\nu \quad (4)$$

Where  $\xi^\mu$  is a vector on a locally flat frame. To ensure the vector is on a flat frame, one must impose the following condition:

$$\frac{\partial^2}{\partial t^2} \xi^\mu(t) \equiv 0 \quad (5)$$

Since a vector is on a flat frame, it should be in a free-falling motion, so its acceleration should be trivial. On a locally flat frame, the metric also becomes flat Euclidean metric

$$g_{\mu\nu} = 1_{\mu\nu} \quad (6)$$

## B.2 COVARIANCE

The vector should be transformed in the same manner in any coordinate frame. However, if the space is no longer flat, the ordinary derivative no longer guarantees it. Let us consider a derivative of a vector in a general curved space.

$$\partial_\mu \rightarrow \partial'_\mu = \frac{\partial x^\mu}{\partial x'^\nu} \partial_\nu \quad (7)$$

Where  $\partial_\mu = \frac{\partial}{\partial x^\mu}$ , then the vector transformation can be written as follows:

$$\partial_\nu X^\mu \rightarrow \partial'_\nu X'^\mu = \frac{\partial x^\lambda}{\partial x'^\nu} \frac{\partial}{\partial x^\lambda} \left( \frac{\partial x'^\mu}{\partial x^\rho} V^\rho \right) \quad (8)$$

$$= \frac{\partial x'^\mu}{\partial x^\lambda} \left( \frac{\partial x'^\rho}{\partial x^\nu} \partial^\lambda V^\rho + \frac{\partial^2 x'^\mu}{\partial x^\lambda \partial x^\rho} V^\rho \right) \quad (9)$$

As it is shown above, a transformation of a vector on a curved space with an ordinary derivative is no longer covariant. Thus, one must impose an additional factor to make it covariant, namely an Affine connection. With this factor, one can define a covariant derivative, replacing an ordinary one.

$$\nabla_\mu = \partial_\mu + \Gamma_{\mu\nu}^\lambda \quad (10)$$

By requiring a covariance condition on the covariant derivative,

$$\nabla_\lambda \rightarrow \nabla'_\lambda V'^\mu = \frac{\partial x^\rho}{\partial x'^\nu} \frac{\partial x'^\mu}{\partial x^\nu} \nabla_\rho V^\nu \quad (11)$$

Then one can induce the explicit form of a connection.

$$\nabla_\mu V^\nu = \partial_\mu V^\nu + \Gamma_{\mu\lambda}^\nu V^\lambda \quad (12)$$

Under coordinate transformation,

$$\frac{\partial}{\partial x'^\mu} \left( \frac{\partial x'^\nu}{\partial x^\lambda} V^\lambda \right) + \Gamma_{\mu\sigma}^{\nu'} V'^\sigma = \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x'^\nu}{\partial x^\lambda} \partial_\rho V^\lambda + \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\lambda} V^\lambda + \Gamma_{\mu\sigma}^{\nu'} V'^\sigma \quad (13)$$

Here, to make the derivative of a vector covariant, the following equation must hold:

$$\frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\lambda} V^\lambda + \Gamma_{\mu\sigma}^{\nu'} V'^\sigma = \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x'^\nu}{\partial x^\lambda} \Gamma_{\rho\sigma}^\lambda V^\sigma \quad (14)$$

Which is

$$\Gamma_{\mu\sigma}^{\nu'} \left( \frac{\partial x'^\sigma}{\partial x^\tau} V^\tau \right) = \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial'^\nu}{\partial x^\lambda} \Gamma_{\rho\sigma}^\lambda V^\sigma - \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\lambda} V^\lambda \quad (15)$$

$$\Gamma_{\mu\kappa}^{\nu'} V^\tau = \frac{\partial x^\rho}{\partial x'^\kappa} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x'^\nu}{\partial x^\lambda} \Gamma_{\rho\sigma}^\lambda V^\sigma - \frac{\partial x^\tau}{\partial x'^\kappa} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\lambda} V^\lambda \quad (16)$$

This leads us to the explicit form of how the Christoffel symbol transforms under coordinate changes.

$$\Gamma_{\mu\kappa}^{\nu'} = \frac{\partial x^\tau}{\partial x'^\kappa} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x'^\nu}{\partial x^\lambda} \Gamma_{\rho\tau}^\lambda - \frac{\partial x^\tau}{\partial x'^\kappa} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\tau} \quad (17)$$

Since the Kronecker delta is a constant matrix, it is obvious that the derivative of the delta should be trivial. Then one can apply the chain rule to the delta and find the following relation, which can simplify the above transformation rule.

$$\frac{\partial}{\partial x'^\mu} \delta_\kappa^\nu = \frac{\partial}{\partial x'^\mu} \frac{\partial x'^\nu}{\partial x'^\kappa} = \frac{\partial}{\partial x'^\mu} \left( \frac{\partial x^\tau}{\partial x'^\kappa} \frac{\partial x'^\nu}{\partial x^\tau} \right) = 0 = \frac{\partial x^\tau}{\partial x'^\kappa} \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial^2 x'^\nu}{\partial x^\rho \partial x^\tau} + \frac{\partial x'^\nu}{\partial x^\tau} \frac{\partial x'^\nu}{\partial x^\tau} \frac{\partial^2 x^\tau}{\partial x'^\mu \partial x'^\rho} \quad (18)$$

Finally, the transformation rule for a Christoffel symbol is as follows:

$$\Gamma'^{\nu}_{\mu\kappa} = \frac{\partial x^{\tau}}{\partial x'^{\kappa}} \frac{\partial x^{\rho}}{\partial x'^{\mu}} \frac{\partial x'^{\nu}}{\partial x^{\lambda}} \Gamma^{\lambda}_{\rho\tau} + \frac{\partial x'^{\nu}}{\partial x^{\tau}} \frac{\partial^2 x^{\tau}}{\partial x'^{\mu} \partial x'^{\rho}} \quad (19)$$

By the same logic, one can easily find out how covariant derivatives act on forms.

$$\nabla_{\mu} V_{\nu} = \partial_{\mu} V_{\nu} - \Gamma^{\lambda}_{\mu\nu} V_{\lambda} \quad (20)$$

### B.3 EXPLICIT FORM OF CHRISTOFFEL SYMBOL

The metric is a ruler of a given geometry; it should not vary under position on a coordinate. The Euclidean is trivial to see since the metric on Euclidean space is mere  $\delta_{\mu\nu}$ , which is a constant matrix.

$$\frac{\partial}{\partial x^{\lambda}} \delta_{\mu\nu} = 0 \quad (21)$$

However, in the curved case, the above statement should also hold to interpret the metric as a ruler, yet the statement does not hold for an ordinary derivative. There, the covariant derivative kicks in to replace an ordinary derivative instead. By taking covariant derivative to the curved metric, the term diminishes.

$$\nabla_{\lambda} g_{\mu\nu} = 0 \quad (22)$$

One can express this in terms of a flat metric with a diffeomorphism transformation factor.

$$g_{\mu\nu}(x) = \frac{\partial \xi^{\lambda}}{\partial x^{\mu}} \frac{\partial \xi^{\rho}}{\partial x^{\nu}} \delta_{\lambda\rho}(\xi) \quad (23)$$

If we take a derivative of  $x$  on both sides, the above equation becomes:

$$\frac{\partial}{\partial x^{\sigma}} g_{\mu\nu}(x) = \frac{\partial^2 x^{\lambda}}{\partial x^{\sigma} \partial x^{\mu}} \frac{\xi^{\rho}}{\partial x^{\nu}} \delta_{\lambda\rho} + \frac{\partial^2 \xi^{\rho}}{\partial x^{\sigma} \partial x^{\nu}} \frac{\partial \xi^{\lambda}}{\partial x^{\mu}} \delta_{\lambda\rho} \quad (24)$$

$$= \frac{\partial^2 \xi^{\rho}}{\partial x^{\sigma} \partial x^{\nu}} \frac{\partial x^{\tau}}{\partial \xi^{\rho}} \frac{\partial \xi^{\lambda}}{\partial x^{\mu}} \delta_{\lambda\rho} + \frac{\partial^2 \xi^{\lambda}}{\partial x^{\sigma} \partial x^{\mu}} \frac{\partial x^{\tau}}{\partial \xi^{\lambda}} \frac{\partial \xi^{\rho}}{\partial x^{\nu}} \delta_{\lambda\rho} \quad (25)$$

$$= \frac{\partial^2 \xi^{\rho}}{\partial x^{\sigma} \partial x^{\nu}} \frac{\partial x^{\tau}}{\partial \xi^{\rho}} g_{\mu\tau} + \frac{\partial^2 \xi^{\lambda}}{\partial x^{\sigma} \partial x^{\mu}} \frac{\partial x^{\tau}}{\partial \xi^{\lambda}} g_{\tau\nu} \quad (26)$$

From eq 22, one can easily find out the specific form of the Christoffel symbol in terms of derivatives of curved and flat coordinates.

$$\frac{\partial}{\partial x^{\sigma}} g_{\mu\nu} = \Gamma^{\tau}_{\sigma\mu} g_{\tau\nu} + \Gamma^{\tau}_{\nu\sigma} g_{\mu\tau} \quad (27)$$

$$\Gamma^{\tau}_{\sigma\mu} = \frac{\partial^2 \xi^{\lambda}}{\partial x^{\sigma} \partial x^{\mu}} \frac{\partial x^{\tau}}{\partial \xi^{\lambda}}(x) \quad (28)$$

Since the metric should always be symmetric, the lower indices of the Christoffel symbol should also be symmetric. It is called a torsion-free condition. Furthermore, by utilizing a simple mathematical trick, one can obtain the Christoffel symbol in terms of the metric  $g_{\mu\nu}$ .

$$\frac{\partial}{\partial x^{\sigma}} g_{\mu\nu} = \Gamma^{\tau}_{\sigma\mu} g_{\tau\nu} + \Gamma^{\tau}_{\sigma\nu} g_{\mu\tau} \quad (29)$$

$$\frac{\partial}{\partial x^{\mu}} g_{\nu\sigma} = \Gamma^{\tau}_{\mu\nu} g_{\tau\sigma} + \Gamma^{\tau}_{\mu\sigma} g_{\nu\tau} \quad (30)$$

$$\frac{\partial}{\partial x^{\nu}} g_{\sigma\mu} = \Gamma^{\tau}_{\nu\sigma} g_{\tau\mu} + \Gamma^{\tau}_{\nu\mu} g_{\sigma\tau} \quad (31)$$

Adding the first two equations and subtracting the last one leads to

$$\Gamma^{\lambda}_{\mu\nu} = \frac{1}{2} g^{\lambda\rho} \left( \frac{\partial}{\partial x^{\mu}} g_{\nu\rho} + \frac{\partial}{\partial x^{\nu}} g_{\rho\mu} - \frac{\partial}{\partial x^{\rho}} g_{\mu\nu} \right) \quad (32)$$

#### B.4 GEODESIC EQUATIONS

The shortest path between two points is simple in flat space. However, in curved space, the notion becomes rather complicated. The shortest path in a curved space is defined as a geodesic. There are several ways to induce a geodesic equation. One is by requiring a free-falling condition.

$$\frac{\partial^2 \xi^\mu(\tau)}{\partial \tau^2} = 0 \quad (33)$$

By diffeomorphism, one can transform a coordinate into an arbitrary coordinate  $x$ .

$$0 = \frac{\partial}{\partial \tau} \left( \frac{\partial \xi^\mu}{\partial x^\nu} \frac{\partial x^\nu}{\partial \tau} \right) = \frac{\partial \xi^\mu}{\partial x^\nu} \frac{\partial^2 x^\nu}{\partial \tau^2} + \frac{\partial^2 \xi^\mu}{\partial x^\lambda \partial x^\nu} \frac{\partial x^\lambda}{\partial \tau} \frac{\partial x^\nu}{\partial \tau} \quad (34)$$

$$\frac{\partial^2 x^\rho}{\partial \tau^2} + \frac{\partial^2 \xi^\mu}{\partial x^\lambda \partial x^\nu} \frac{\partial x^\rho}{\partial \xi^\mu} \frac{\partial x^\lambda}{\partial \tau} \frac{\partial x^\nu}{\partial \tau} = \frac{\partial^2 x^\rho}{\partial \tau^2} + \Gamma^\rho_{\lambda\nu} \frac{\partial x^\lambda}{\partial \tau} \frac{\partial x^\nu}{\partial \tau} = 0 \quad (35)$$

Another way to derive the equation is by finding the minimum value of the distance in curved space.

$$S = \int \sqrt{g_{\mu\nu} \frac{dx^\mu}{d\tau} \frac{dx^\nu}{d\tau}} d\tau \quad (36)$$

By varying the above equation and requiring it to be 0, one can compute its minimum value, and after tedious calculation, the geodesic equation can be obtained.

## C BASE GRAPH NEURAL NETWORK MODEL

In general, molecule is represented in a graph form. Therefore, in order to handle molecule dataset, it is inevitable to utilize graph neural networks. We chose directional message passing network (DMPNN) (Yang et al., 2019) for our backbone, since it outperforms other GNN architectures in molecular domain. Given a graph, DMPNN initializes the hidden state of each edge  $(i, j)$  based on its edge feature  $E_{ij}$  with node feature  $X_i$ . At each step  $t$ , directional edge summarizes incident edges as a message  $m_{ij}^{t+1}$  and updates its hidden state to  $h_{ij}^{t+1}$ .

$$m_{ij}^{t+1} = \sum_{k \in \mathcal{N}(i) \setminus j} h_{ki}^t \quad (37)$$

$$h_{ij}^{t+1} = \text{ReLU}(h_{ij}^0 + W_e m_{ij}^{t+1}) \quad (38)$$

Where  $\mathcal{N}(i)$  denotes the set of neighboring nodes and  $W_e$  a learnable weight. The hidden states of nodes are updated by aggregating the hidden states of incident edges into message  $m_i^{t+1}$ , and passing its concatenation with the node feature  $X_i$  into a linear layer followed by ReLU non-linearity

$$m_i^{t+1} = \sum_{j \in \mathcal{N}(i)} h_{ij}^t \quad (39)$$

$$h_i^{t+1} = \text{ReLU}(W_n \text{concat}(X_i, m_i^{t+1})) \quad (40)$$

Similarly,  $W_n$  denotes a learnable weight. Assuming DMPNN runs for  $T$  timesteps, we use  $(X_{out}, E_{out}) = \text{GNN}(A, X, E)$  to denote the output representation matrices containing hidden states of all nodes and edges, respectively (i.e.,  $X_{out,i} = h_i^T$  and  $E_{out,ij} = h_{ij}^T$ ).

For graph-level prediction, the node representations after the final GNN layer are typically summed to obtain a single graph representation  $h_G = \sum_i h_i$ , which is then passed to a FFN prediction layer.

## D ARCHITECTURE AND HYPERPARAMETERS

Detailed steps of training *GATE* is described in Algorithm 1. The architecture of our model is composed of five distinct networks and their parameter sizes are depicted in Table 1. As illustrated in Figure ??, one embedding network is shared across tasks, and encoder, transfer, inverse transfer, and head network exists for each task. The embedding network *embedd*( $\cdot$ ) has the DMPNN architecture

---

**Algorithm 1** *GATE*

---

```
1: Initialize encoder network  $f_e$ , transfer network  $f_t$ , inverse transfer network  $f_i$ , head network  $f_h$ 
   with random parameters  $\theta$ 
2:
3: for epoch  $i = 1, 2, \dots, n$  do
4:   for each  $t \in Tasks$  do
5:     for each batch  $\mathbf{b} = (x^t, y^t) \in \text{dataset } D$  do
6:        $a^t \leftarrow \text{embedd}(x^t)$ 
7:        $\{\bar{a}^t\} \leftarrow \text{perturb}(a^t)$ 
8:
9:        $z^t \leftarrow f_e^t(a^t)$ 
10:       $m^t \leftarrow f_t^t(z^t)$ 
11:       $\{\bar{z}^t\} \leftarrow f_e^t(\{\bar{a}^t\})$ 
12:       $\{\bar{m}^t\} \leftarrow f_t^t(\{\bar{z}^t\})$ 
13:
14:       $L_{reg} \leftarrow \text{MSELoss}(y^t, f_h^t(z^t))$ 
15:       $L_{auto} \leftarrow \text{MSELoss}(f_i^t(m^t), z^t)$ 
16:
17:      for each  $s \in Subtasks$  do
18:         $z^s \leftarrow f_e^s(a^t)$ 
19:         $m^s \leftarrow f_t^s(z^s)$ 
20:         $\{\bar{z}^s\} \leftarrow f_e^s(\{\bar{a}^s\})$ 
21:         $\{\bar{m}^s\} \leftarrow f_t^s(\{\bar{z}^s\})$ 
22:
23:         $L_{map} \leftarrow L_{map} + \text{MSELoss}(y^t, f_h^t \circ f_i^t(m^s))$ 
24:         $L_{cons} \leftarrow L_{cons} + \text{MSELoss}(\{\bar{m}^t\}, m^s)$ 
25:         $L_{dist} \leftarrow L_{dist} + \text{MSELoss}(m^t - \{\bar{m}^t\}, m^s - \{\bar{m}^s\})$ 
26:      end for
27:
28:      Compute  $L_{total} = L_{reg} + \alpha L_{auto} + \beta L_{map} + \gamma L_{cons} + \delta L_{dist}$ 
29:      Update  $\theta$  using  $L_{total}$ 
30:    end for
31:  end for
32: end for
```

---

with depth 2 and converts the input molecule representation  $x$  into a new representation  $a$  in a common embedding space. We apply perturbation  $\text{perturb}(\cdot)$  to  $a$  for a number of perturbations, which is set to 10 in this paper. All of the perturbed representation  $\{\bar{a}\}$  along with  $a$  are then fed into the encoder network. The encoder network is composed of backbone network and bottleneck network. Backbone network has the DMPNN architecture with depth 2 and the bottleneck network has an autoencoder structure with MLP layers. The output from the encoder  $f_e(a)$  becomes the input to the transfer network and head network. The output of transfer network  $f_t(z)$ , notated as  $m$ , is used to calculate consistency loss and distance loss. It is also fed into inverse transfer network, so that the output from inverse transfer network  $f_i(m)$  can be used to calculate autoencoder loss. The output from head network  $f_h \circ f_i(m)$  is used to calculate regression loss and mapping loss. We trained 600 epochs with batch size 512 while using AdamW (Loshchilov & Hutter, 2017) for optimization with learning rate  $5e-5$ . The hyperparameters for  $\alpha, \beta, \gamma, \delta$  are 1, 1, 1, 1 respectively.

Table 1: Network parameters

network	layer	input, output size	hidden size	dropout
backbone	DMPNN	[134,149], 100	200	0
bottleneck	MLP layer	100, 50	50	0
transfer	MLP layer	50, 50	100,100,100	0.2
inverse transfer	MLP layer	50, 50	100,100,100	0.2
head	MLP layer	50, 1	25,12	0.2

Table 2: Hyperparameters

learning rate	0.00005
optimizer	AdamW
batch size	512
epoch	600
# of perturbation	10
$\alpha, \beta, \gamma, \delta$	1, 1, 1, 1

## E DETAILED EXPLANATION OF DATASETS AND EXPERIMENTAL SETUPS

### E.1 DATASETS

Table 3: Detailed information about the datasets.

name	acronym	source	count	mean	std
Abraham Descriptor S	AS	Ochem	1925	1.05	0.68
Boiling Point	BP	Pubchem	7139	198.99	108.88
Collision Cross Section	CCS	Pubchem	4006	205.06	57.84
Critical Temperature	CT	Ochem	242	626.04	120.96
Dielectric Constant	DK	Ochem	1007	0.80	0.41
Density	DS	Pubchem	3079	1.07	0.29
Enthalpy of Fusion	EF	Ochem	2188	1.32	0.32
Ionization Potential	IP	Pubchem	272	10.00	1.63
Kovats Retention Index	KRI	Pubchem	73507	2071.20	719.34
Log P	LP	Pubchem	28268	11.17	9.89
Polarizability	POL	CCCB	241	0.84	0.26
Surface Tension	ST	Pubchem	379	29.01	10.36
Viscosity	VS	Pubchem	294	0.47	0.87
Heat of Vaporization	HV	Pubchem	525	43.77	18.08

We used 14 different molecular property datasets from three different open databases, described in Table 3 and below explanations for evaluation of the *GATE*. Before the training process, the data were purified to exclude data with incorrectly specified units, typos, and extreme measurement environments. All datasets were normalized by mean and standard deviation before the training process. We selected 23 pairs of source and target tasks among the 14 datasets, considering the number of data points in each dataset. We also tried to select task pairs with diversity in correlation as shown in the Figure 1 for a fair and unbiased examination. Hereby, we explicitly describe the physical meaning of each dataset.

- **AS** : The solute dipolarity/polarizability.
- **BP** : The temperature at which this compound changes state from liquid to gas at a given atmospheric pressure.
- **CCS** : The effective area for the interaction between an individual ion and the neutral gas through which it is traveling.
- **CT** : The temperature when no gas can become liquid no matter how high the pressure is.
- **DK** : The ratio of the electric permeability of the material to the electric permeability of free space.
- **DS** : The mass of a unit volume of a compound.
- **EF** : The change in enthalpy resulting from the addition or removal of heat from 1 mole of a substance to change its state from a solid to a liquid.

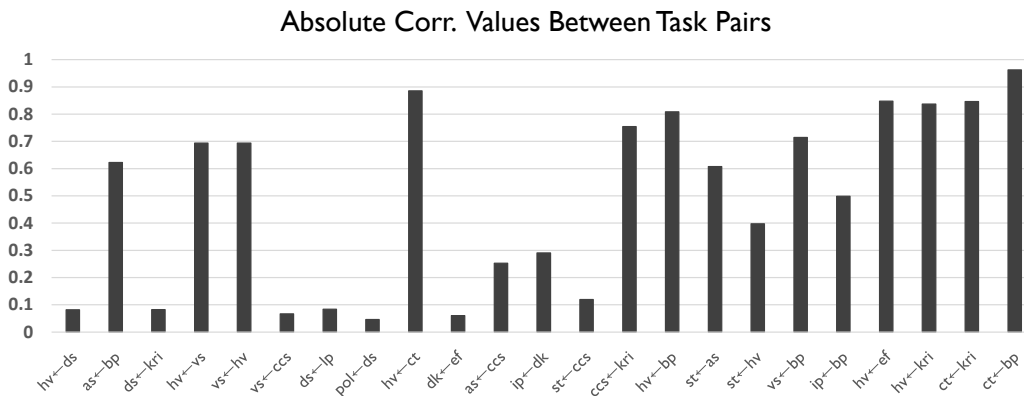


Figure 1: Pearson correlation between overlapping data points in target dataset and source dataset.

- **IP** : The amount of energy required to remove an electron from an isolated atom or molecule.
- **KRI** : The rate at which a compound is processed through a gas chromatography column.
- **LP** : Logarithmic form of the ratio of concentrations of a compound in a mixture of octanol and water at equilibrium.
- **POL** : The tendency of matter, when subjected to an electric field, to acquire an electric dipole moment in proportion to that applied field.
- **ST** : The property of the surface of a liquid that allows it to resist an external force
- **VS** : A measure of a fluid’s resistance to flow.
- **HV** : The quantity of heat that must be absorbed if a certain quantity of liquid is vaporized at a constant temperature.

## E.2 EXPERIMENTAL SETUPS

For evaluation of the *GATE*, we compared the performance of six baseline methods, including STL, MTL, KD, global structure preserving loss based KD (GSP-KD), and transfer learning (retrain all or head network only). All of the baselines share the same base architecture, with a few different details according to methods. The MTL shares parameters of backbone and bottleneck for given two tasks, and only head networks are separated. In the case of the KD, latent vectors from the bottleneck are used as labels for the distillation, and the distillation loss ratio is set to 0.1. Graph Contrastive Representation Distillation (G-CRD) contains contrastive loss as well as the GSP loss (Joshi et al., 2022). However, we only adopt GSP loss since the contrastive loss term is not applicable for regression tasks. For the GSP-KD, node features from the last layer of the backbone network are used to calculate pairwise distances, which are the labels of the distillation process. The loss ratio of the distillation process of the GSP is also set to 0.1. The maximum epoch is set to be 600, and the best models are selected by early stopping.

## F EXPERIMENTAL RESULTS

We express explicit test results in this section. A total of 23 task pairs from 14 distinct datasets were tested thoroughly with seven different models. Four tables are depicted to show the full experimental results. The best result is emphasized by bold and underlined on each individual result, and the second-best result is underlined. The *GATE* outperforms other conventional methods by a noticeable margin. In a random split setup, the *GATE* wins 52.17% out of total tasks, and for up to second, the *GATE* wins 78.26% out of total. In scaffold setup, the *GATE* wins 56.52% and 78.26% respectively.

Table 4: Random Split Result (part 1)

Tasks	GATE		STL		MTL		KD	
	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	STD
hv $\leftarrow$ ds	<b><u>0.9221</u></b>	0.0612	0.9574	0.0519	0.9782	0.0782	1.3726	0.2930
as $\leftarrow$ bp	0.4583	0.0193	0.5125	0.0085	<u>0.4370</u>	0.0119	0.5426	0.0335
ds $\leftarrow$ kri	<b><u>0.4145</u></b>	0.0172	0.4154	0.0045	0.4172	0.0102	0.4403	0.0119
hv $\leftarrow$ vs	<b><u>0.9116</u></b>	0.0522	0.9574	0.0519	0.9700	0.1052	1.1995	0.1419
vs $\leftarrow$ hv	<b><u>0.5471</u></b>	0.0719	0.5947	0.0357	<u>0.5535</u>	0.0353	0.5878	0.0264
st $\leftarrow$ as	<b><u>0.6689</u></b>	0.0413	<u>0.9902</u>	0.0729	1.0272	0.0244	1.1601	0.0396
ds $\leftarrow$ lp	<b><u>0.4046</u></b>	0.0142	0.4154	0.0045	0.4133	0.0135	0.4378	0.0086
pol $\leftarrow$ ds	0.3431	0.0475	0.3460	0.0291	0.4367	0.1213	<u>0.3089</u>	0.0270
vs $\leftarrow$ bp	<b><u>0.4457</u></b>	0.0151	0.5947	0.0357	<u>0.4516</u>	0.0366	0.6076	0.0241
dk $\leftarrow$ ef	0.4331	0.0140	0.4331	0.0358	0.4498	0.0126	<b><u>0.3852</u></b>	0.0238
as $\leftarrow$ ccs	<b><u>0.4648</u></b>	0.0139	0.5125	0.0085	0.4677	0.0220	0.5364	0.0211
ct $\leftarrow$ bp	0.1742	0.0034	0.2549	0.1247	<u>0.1707</u>	0.0132	<u>0.1690</u>	0.0079
st $\leftarrow$ ccs	<b><u>0.9546</u></b>	0.0452	0.9902	0.0729	1.0361	0.0737	1.1731	0.0730
ccs $\leftarrow$ kri	<u>0.2476</u>	0.0034	0.2936	0.0110	0.2524	0.0042	0.2622	0.0117
hv $\leftarrow$ bp	<b><u>0.7251</u></b>	0.0581	0.9574	0.0519	<u>0.7550</u>	0.0432	1.1983	0.1815
vs $\leftarrow$ ccs	<u>0.5233</u>	0.0323	0.5947	0.0357	0.5792	0.0228	0.6027	0.0127
st $\leftarrow$ hv	<u>0.7647</u>	0.0622	0.9902	0.0729	<b><u>0.7179</u></b>	0.0259	1.1270	0.0184
hv $\leftarrow$ ct	<u>0.9399</u>	0.0896	0.9574	0.0519	1.1118	0.1633	1.5114	0.1845
ip $\leftarrow$ bp	<u>0.5476</u>	0.0642	0.6695	0.0660	0.6067	0.0345	0.5624	0.0273
hv $\leftarrow$ ef	<b><u>0.6131</u></b>	0.0966	0.9574	0.0519	0.8296	0.0999	1.3659	0.2587
hv $\leftarrow$ kri	<b><u>0.5410</u></b>	0.0732	0.9574	0.0519	<u>0.8631</u>	0.0354	1.3739	0.2487
ct $\leftarrow$ kri	<u>0.1658</u>	0.0136	0.2549	0.1247	0.1716	0.0090	<b><u>0.1586</u></b>	0.0102
ip $\leftarrow$ dk	<u>0.6510</u>	0.0381	0.6695	0.0660	0.7083	0.0226	<u>0.5508</u>	0.0100
mean	<b><u>0.5592</u></b>	0.0271	0.6642	0.0320	0.6263	0.0414	0.7667	0.0908
	Count	Ratio	Count	Ratio	Count	Ratio	Count	Ratio
1st	12	52.17%	0	0.00%	1	4.35%	2	8.70%
2nd	18	78.26%	1	4.35%	7	30.43%	5	21.74%



Table 5: Random Split Result (part 2)

Tasks	GSP-KD		Transfer Retrain All		Transfer Retrain Head	
	RMSE	STD	RMSE	STD	RMSE	STD
hv $\leftarrow$ ds	<u>0.9321</u>	0.0487	1.0428	0.1165	1.1166	0.0024
as $\leftarrow$ bp	0.5315	0.0151	<b>0.4325</b>	0.0104	0.7712	0.0105
ds $\leftarrow$ kri	<u>0.4147</u>	0.0063	0.4414	0.0154	0.8842	0.0049
hv $\leftarrow$ vs	<u>0.9154</u>	0.0130	0.9937	0.0821	1.0091	0.0181
vs $\leftarrow$ hv	0.5619	0.0223	0.5712	0.0232	0.7215	0.0392
st $\leftarrow$ as	0.9938	0.0141	1.1296	0.1302	1.0045	0.0220
ds $\leftarrow$ lp	<u>0.4106</u>	0.0077	0.4280	0.0136	0.9111	0.0022
pol $\leftarrow$ ds	<b>0.2603</b>	0.0270	0.3741	0.0303	0.9060	0.0141
vs $\leftarrow$ bp	0.5932	0.0097	0.5445	0.0239	0.7220	0.0645
dk $\leftarrow$ ef	0.4230	0.0133	<u>0.3936</u>	0.0164	0.9380	0.0026
as $\leftarrow$ ccs	0.5457	0.0150	0.4741	0.0148	0.9935	0.0033
ct $\leftarrow$ bp	0.2018	0.0093	<b>0.1563</b>	0.0044	0.6847	0.0186
st $\leftarrow$ ccs	<u>0.9595</u>	0.0405	1.1334	0.0687	1.1039	0.0046
ccs $\leftarrow$ kri	0.2698	0.0095	<b>0.2273</b>	0.0016	0.6166	0.0567
hv $\leftarrow$ bp	0.9051	0.0571	0.8267	0.0417	0.8829	0.0499
vs $\leftarrow$ ccs	0.5269	0.0167	<b>0.4868</b>	0.0119	0.8684	0.0116
st $\leftarrow$ hv	0.9618	0.0086	1.0290	0.0945	1.0102	0.0138
hv $\leftarrow$ ct	<b>0.9207</b>	0.0112	1.2072	0.0460	1.0302	0.0186
ip $\leftarrow$ bp	<b>0.4631</b>	0.0037	0.9816	0.2334	0.8732	0.0293
hv $\leftarrow$ ef	<u>0.8112</u>	0.0463	1.0818	0.1021	0.9616	0.0478
hv $\leftarrow$ kri	0.9191	0.0676	0.9080	0.0510	1.0715	0.0145
ct $\leftarrow$ kri	0.2080	0.0057	0.1661	0.0075	0.8349	0.0279
ip $\leftarrow$ dk	<b>0.5257</b>	0.0192	0.6099	0.0273	1.0336	0.0085
mean	0.6198	0.0212	0.6800	0.0540	0.9108	0.0181
	Count	Ratio	Count	Ratio	Count	Ratio
1st	4	17.39%	4	17.39%	0	0.00%
2nd	10	43.48%	5	21.74%	0	0.00%

Table 6: Scaffold Split Result (part 1)

Tasks	GATE		STL		MTL		KD	
	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	STD
hv $\leftarrow$ ds	0.6939	0.0996	0.6744	0.1079	<u>0.6465</u>	0.0776	<b>0.5920</b>	0.0466
as $\leftarrow$ bp	<b>1.0495</b>	0.0256	1.2828	0.0724	1.1677	0.1068	1.3580	0.0136
ds $\leftarrow$ kri	<b>0.4395</b>	0.0108	0.4477	0.0052	0.4849	0.0061	0.5409	0.0480
hv $\leftarrow$ vs	0.7174	0.0796	<u>0.6744</u>	0.1079	0.9954	0.2059	0.8948	0.2294
vs $\leftarrow$ hv	<b>0.6120</b>	0.0639	<u>0.9816</u>	0.1267	0.8535	0.0558	1.2597	0.3638
st $\leftarrow$ as	<b>0.7540</b>	0.0660	<u>0.8041</u>	0.1062	1.0254	0.0251	1.7083	0.1608
ds $\leftarrow$ lp	<b>0.4049</b>	0.0102	<u>0.4477</u>	0.0052	0.4517	0.0184	0.5221	0.0328
pol $\leftarrow$ ds	<u>0.9040</u>	0.0852	0.9604	0.1056	1.4198	0.0796	1.3309	0.1998
vs $\leftarrow$ bp	<u>0.6121</u>	0.0297	0.9816	0.1267	<b>0.5686</b>	0.0276	0.9371	0.2386
dk $\leftarrow$ ef	0.7122	0.0545	0.7028	0.0391	<u>0.6549</u>	0.0210	0.8189	0.0462
as $\leftarrow$ ccs	1.1313	0.0496	1.2828	0.0724	<b>1.1197</b>	0.0558	1.3773	0.0781
ct $\leftarrow$ bp	<b>0.3883</b>	0.0203	1.4436	0.1150	<u>0.4359</u>	0.0126	1.2459	0.1199
st $\leftarrow$ ccs	<b>0.7281</b>	0.0586	0.8041	0.1062	0.9905	0.0737	1.5402	0.1418
ccs $\leftarrow$ kri	<b>0.5292</b>	0.0094	0.5489	0.0107	<u>0.5297</u>	0.0083	0.5534	0.0190
hv $\leftarrow$ bp	<u>0.4821</u>	0.0132	0.6744	0.1079	<b>0.4668</b>	0.0169	0.6271	0.0868
vs $\leftarrow$ ccs	<b>0.6126</b>	0.0671	0.9816	0.1267	0.8186	0.0790	1.3034	0.5354
st $\leftarrow$ hv	<b>0.7209</b>	0.0412	0.8041	0.1062	<u>0.7237</u>	0.0276	1.5256	0.1906
hv $\leftarrow$ ct	<u>0.6579</u>	0.0678	0.6744	0.1079	0.6633	0.0660	0.7925	0.2694
ip $\leftarrow$ bp	0.4668	0.0179	0.5780	0.1475	0.5540	0.0587	<b>0.4205</b>	0.0240
hv $\leftarrow$ ef	<u>0.6406</u>	0.0335	0.6744	0.1079	0.7879	0.0643	0.6773	0.1553
hv $\leftarrow$ kri	<b>0.5084</b>	0.0264	0.6744	0.1079	0.6204	0.0269	0.6710	0.1524
ct $\leftarrow$ kri	<b>0.3902</b>	0.0140	1.4436	0.1150	<u>0.5173</u>	0.0927	1.3392	0.1076
ip $\leftarrow$ dk	<b>0.4335</b>	0.0119	0.5780	0.1475	<u>0.5335</u>	0.1016	0.4975	0.0769
mean	<b>0.6343</b>	0.0270	0.8313	0.0408	<u>0.7404</u>	0.0441	0.9797	0.1218
	Count	Ratio	Count	Ratio	Count	Ratio	Count	Ratio
1st	13	56.52%	0	0.00%	3	13.04%	2	8.70%
2nd	18	78.26%	3	13.04%	9	39.13%	2	8.70%

Table 7: Scaffold Split Result (part 2)

Tasks	GSP-KD		Transfer Retrain All		Transfer Retrain Head	
	RMSE	STD	RMSE	STD	RMSE	STD
hv $\leftarrow$ ds	0.7606	0.0810	0.8659	0.0788	0.9584	0.0339
as $\leftarrow$ bp	1.2340	0.0294	1.1478	0.0264	<u>1.0935</u>	0.0079
ds $\leftarrow$ kri	<u>0.4467</u>	0.0104	0.8753	0.1134	1.0928	0.0482
hv $\leftarrow$ vs	<b><u>0.6536</u></b>	0.0345	0.7520	0.1666	0.7924	0.0595
vs $\leftarrow$ hv	<u>0.6377</u>	0.0253	0.9217	0.1575	0.9179	0.0539
st $\leftarrow$ as	0.9335	0.0954	1.2604	0.0946	1.0780	0.0613
ds $\leftarrow$ lp	0.4685	0.0111	0.4664	0.0121	1.0410	0.0026
pol $\leftarrow$ ds	<b><u>0.8475</u></b>	0.0627	1.0385	0.2146	1.3204	0.0491
vs $\leftarrow$ bp	0.6599	0.0204	1.1532	0.1766	1.0135	0.0820
dk $\leftarrow$ ef	<b><u>0.6353</u></b>	0.0171	0.7417	0.0384	0.7963	0.0071
as $\leftarrow$ ccs	<u>1.1272</u>	0.0778	1.2925	0.0606	1.4530	0.0143
ct $\leftarrow$ bp	1.1837	0.0586	0.5644	0.053	0.9347	0.0316
st $\leftarrow$ ccs	<u>0.7344</u>	0.0187	0.9075	0.0431	1.2596	0.0287
ccs $\leftarrow$ kri	0.5356	0.0115	0.5640	0.0137	0.7904	0.0159
hv $\leftarrow$ bp	0.7403	0.0889	0.6093	0.0422	0.8111	0.0251
vs $\leftarrow$ ccs	0.8027	0.0159	<u>0.7271</u>	0.0828	1.2282	0.0243
st $\leftarrow$ hv	0.7417	0.0206	1.4243	0.0627	1.0047	0.0813
hv $\leftarrow$ ct	<b><u>0.6428</u></b>	0.008	0.9499	0.2579	0.8089	0.0532
ip $\leftarrow$ bp	0.4579	0.0207	<u>0.4419</u>	0.0371	0.9704	0.0399
hv $\leftarrow$ ef	<b><u>0.5862</u></b>	0.0375	1.0003	0.1719	0.9503	0.0307
hv $\leftarrow$ kri	<u>0.5509</u>	0.0252	0.6560	0.0408	0.9998	0.0311
ct $\leftarrow$ kri	1.2358	0.0373	1.1124	0.1265	1.2769	0.0193
ip $\leftarrow$ dk	0.4376	0.0255	0.5248	0.0471	1.0165	0.0521
mean	0.7415	0.0363	0.8694	0.0671	1.0265	0.0217
	Count	Ratio	Count	Ratio	Count	Ratio
1st	5	21.74%	0	0.00%	0	0.00%
2nd	11	47.83%	2	8.70%	1	4.35%

---

## REFERENCES

- Chaitanya K Joshi, Fayao Liu, Xu Xun, Jie Lin, and Chuan Sheng Foo. On representation knowledge distillation for graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Tim Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59, 07 2019. doi: 10.1021/acs.jcim.9b00237.