GEOMETRICALLY ALIGNED TRANSFER ENCODER FOR INDUCTIVE TRANSFER IN REGRESSION TASKS

Sung Moon Ko*, **Sumin Lee***, **Dae-Woong Jeong***, **Woohyung Lim**, **Sehui Han**
LG AI Research
{sungmoon.ko, sumin.lee, dw.jeong, w.lim, hansse.han}@lgresearch.ai

ABSTRACT

Transfer learning is a crucial technique for handling a small amount of data that is potentially related to other abundant data. However, most of the existing methods are focused on classification tasks using images and language datasets. Therefore, in order to expand the transfer learning scheme to regression tasks, we propose a novel transfer technique based on differential geometry, namely the Geometrically Aligned Transfer Encoder (*GATE*). In this method, we interpret the latent vectors from the model to exist on a Riemannian curved manifold. We find a proper diffeomorphism between pairs of tasks to ensure that every arbitrary point maps to a locally flat coordinate in the overlapping region, allowing the transfer of knowledge from the source to the target data. This also serves as an effective regularizer for the model to behave in extrapolation regions. In this article, we demonstrate that *GATE* outperforms conventional methods and exhibits stable behavior in both the latent space and extrapolation regions for various molecular graph datasets.

1 INTRODUCTION

A machine learning model requires abundant data to learn and perform effectively. However, collecting sufficient amounts of data often consumes substantial amounts of energy and time. One of the most practical approaches to addressing this issue is to make use of large datasets correlated with the target data. Transfer learning stands out as one of the potential contenders. Transfer learning focuses on transferring knowledge from source to target data. Recently, there has been immense development in various domains such as languages (Zhuang et al., 2011; Long et al.; Zhuang et al., 2013; 2014; Pan et al., 2020), computer vision (Quattoni et al., 2008; Kulis et al., 2011; Raghu et al., 2019; Yu et al., 2022) and biomedical domain (Wang et al., 2019; Peng et al., 2021).

Despite these achievements, the primary area of concern is mostly limited to classification tasks in the vision and language domains. Moreover, identifying an architecture that effectively addresses regression problems has proven to be challenging. Hence, our primary objective is to develop an algorithm that can be generally applied to regression problems. In this article, our main focus centers on molecular property prediction tasks (Scarselli et al., 2009; Bruna et al., 2013; Duvenaud et al., 2015; Defferrard et al., 2016; Jin et al., 2018; Coley et al., 2019; Ko et al., 2023). Molecular datasets tend to be small in size, have a large number of task types, and are mainly regression. This makes molecular property prediction a good application to test our novel algorithm.

The fundamental concept of our method rests on a geometric interpretation of the encoding space. Given that the encoding process within a model involves intricate non-linear mappings, it is reasonable to regard the encoding space as non-linear and curved. As it is illustrated in Figure 1, we assume a data point in each task is on a manifold, and its collection formulates a coordinate patch that can be interpreted as a task coordinate. Each task coordinate has a chart to map to another correlated task to form a manifold. Additionally, machine learning models are obligated to exhibit continuity and differentiability to facilitate training through backpropagation. This requirement extends to the encoding space, which should ideally possess smoothness properties. These observations collectively suggest the possibility of interpreting the encoding space as a Riemannian manifold.

*These authors contributed equally to this work

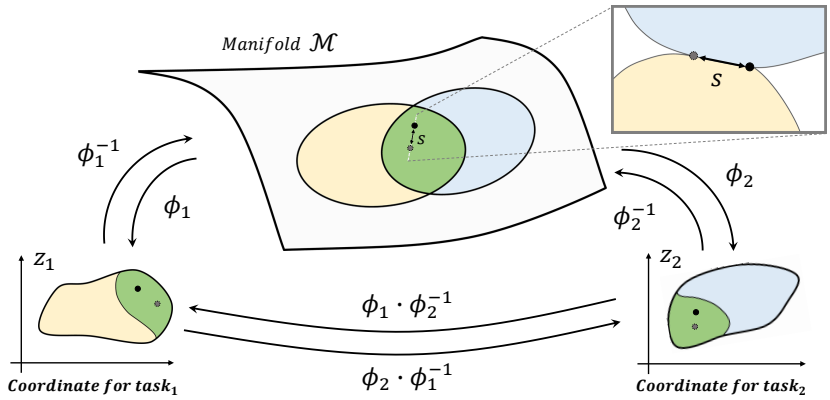

Figure 1: Two different coordinate frames are demonstrated in the figure, with coordinate transformation maps to each individual coordinate choice. One can interpret each coordinate frame as task-specific coordinates and map them with transformation models. An arbitrary point in the overlapping region of the manifold can be transformed from one task coordinate to another by combining mapping functions ϕ. Moreover, by introducing perturbation points, as demonstrated in the figure, one can define the distance between points to match the geometrical shape in the overlapping region.

In general, a curved space is considerably more intricate than a flat space. For instance, when dealing with a curved space, determining the distance between points requires the computation of entire geodesic equations based on a specific metric. Consequently, the straightforward calculation of similarities between data points using Euclidean distance no longer applies, making it challenging to discern meaningful similarities between points. However, when the manifold is regarded as a Riemannian manifold, there are strategies that can help navigate these challenges. One such strategy involves finding diffeomorphisms that allow points to be situated in a locally flat frame (LF). As a result, it becomes feasible to develop models specifically designed to identify appropriate diffeomorphisms. These models aim to confine individual data points to LF within overlapping regions that align with other coordinate patches. This approach simplifies the process of establishing correspondence between similar points across different coordinate systems, as each point is effectively situated within a Euclidean environment on a local scale.

In essence, the act of matching overlapping points on a manifold ensures the alignment of their geometries. This alignment is crucial because the encoding space's inherent geometry in the source data contains relevant information for the target data. By achieving this alignment, the model facilitates the flow of information from the source to the target. In effect, this approach establishes a strong foundation for the model's capabilities and extends its capacity to generalize effectively.

Our main contribution of the article is as follows.

- We design a novel transfer algorithm (*GATE*) based on Riemannian differential geometry.
- The *GATE* is utilizable on regression tasks for inductive transfer learning.
- The *GATE* outperforms conventional methods in various molecular property regressions.
- The *GATE* exhibits stable underlying geometry and robust behavior in extrapolation tasks.

2 Related Works

2.1 Transfer Learning

Given a source domain D^s with source task T^s and a target domain D^t with target task T^t, transfer learning aims to improve the learning of a better mapping function $f^t(\cdot)$ for the target task T^t with the transferable knowledge gained from the D^s and T^s (Pan & Yang, 2009). Depending on the relationship between the source and target domains and tasks, transfer learning can be divided into three categories: inductive, transductive, and unsupervised transfer learning. Our problem setting is inductive transfer learning, where the target task is different from the source task ($T^s \neq T^t$),

regardless of whether the source and target domains are the same (Yang et al., 2020). Additionally, molecular property prediction is a regression task. In transductive transfer learning, studies have been done on regression tasks such as domain adaptation regression (DAR) (Chen et al., 2021). However, recent studies on regression tasks in an inductive transfer learning setting were only found in papers that applied it to a specific domain (Li & Fourches, 2020; Hoffmann et al., 2023).

2.2 Multi-task Learning and Knowledge Distillation

Multi-task learning (MTL) can be viewed as a form of inductive transfer learning that learns multiple related tasks simultaneously. By using a shared representation, the knowledge gained from one task can help the learning of other tasks (Caruana, 1997). MTL and transfer learning both aim to generalize knowledge across different tasks. However, MTL aims to improve performance on a set of target tasks with equal importance, whereas transfer learning only focuses on one target task. Studies using MTL in various domains can be found (Lee & Kim, 2019; Liu et al., 2022).

Knowledge distillation (KD) is a training paradigm where the knowledge from a large teacher model is transferred to a small student model. According to (Gou et al., 2021), distillation techniques can be classified by knowledge types: response-based, feature-based, and related-based knowledge. Response-based knowledge, also called "dark knowledge", uses the output of the last layer, such as logits (Hinton et al., 2015). Feature-based knowledge distillation utilizes the output of intermediate layers, which are feature representations, to guide the student model. Relation-based KD transfers the relationships between features or data samples. For molecular property prediction, (Joshi et al., 2022) introduced feature-based distillation for GNNs by preserving global topology. Studies using cross-modal KD can also be found (Zhang et al., 2022; Zeng et al., 2023).

2.3 Geometrical Deep Learning

Methods that extend deep neural networks to Euclidean and non-Euclidean domains, such as graphs and manifolds, are collectively referred to as geometric deep learning(GDL) (Bronstein et al., 2017). Regarding "how to transfer", transfer learning can be categorized into instance-based, feature-based, model-based, and relation-based transfer learning. A feature-based approach aims to find a feature representation that is effective for both the source and target domains. In the context of domain adaptation, the subspace spanned by the features in the source and target domains is the knowledge that is transferred (Yang et al., 2020). Manifold feature learning can learn tight representations that are invariant across domains. Some studies have used the property of manifold to perform unsupervised domain adaptation. By embedding datasets into Grassmann mainfolds, (Gopalan et al., 2011) obtained intermediate subspaces by sampling points along the geodesic between the source and target subspaces. (Gong et al., 2012) extended on this by learning a geodesic flow kernel (GFK) between domains. (Baktashmotlagh et al., 2014) embedded the probability distributions on a Riemannian manifold and used a Hellinger distance to approximate the geodesic distance. (Luo et al., 2020) proposed a Riemannain manifold embedding and alignment framework that used intra-class similarity and manifold metric alignment loss to achieve discriminability and transferability, respectively. However, to our best knowledge, we could not find geometrical deep learning approaches for inductive transfer learning in regression tasks.

3 Geometrically Aligned Transfer Encoder

The latent vector is believed to capture the essence of information for a given task. Therefore, one may consider if two different but correlated tasks may share a similar underlying geometry of latent spaces. Hence, if the latent spaces of given tasks can be aligned smoothly, mutual information will flow through the latent spaces from one another. This will lead to the superior performance of a target task not only in interpolation but also in extrapolation cases, which is the basic strategy of the *GATE*. However, deep learning models are designed to mimic non-linear functions by excavating information from the training dataset. The underlying geometry of models should inherently exhibit non-linearity and non-trivial curvature, as well as latent spaces, which are induced by non-linear models. This makes it challenging to match the underlying geometry of latent spaces.

There are several assumptions required to put geometrical concept into practice: 1) A pair of source and target tasks should be correlated; 2) a pair of tasks should have an overlapping region; 3) a

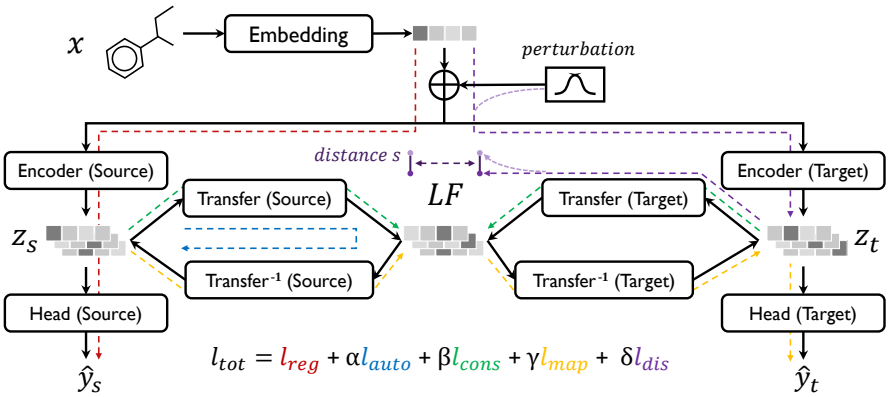

Figure 2: Schematic diagram for the *GATE* algorithm

mathematically well-defined underlying geometry is preferred, such as a Riemannian. Molecular datasets provide a perfect proving ground since they satisfy the first two assumptions, and with some proper constraints, the third assumption can also be established. First, there exist numerous sets of correlated tasks proven by scientific research. Also, the majority of molecules possess values for multiple properties, which ensures a large area of overlapping regions across a pair of tasks. Finally, by imposing the mathematical characters of Riemannian geometry as constraints, the underlying geometry can be interpreted as a Riemannian manifold. Moreover, every molecule can be expressed in a single corresponding universal string in, for instance, the SMILES format(Weininger, 1988).

3.1 Riemannian Geometry

In Figure 2 we first take an input SMILES and embed it into the corresponding vector. After embedding, latent space is formulated by encoders, which consist of DMPNN(Yang et al., 2019) and MLP layers. The latent vector is fed into task-corresponding heads for inference properties. Here we utilize MSE for basic regression loss in the training scheme as follows:

$$l_{\text{reg}} = \frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

Where N, y_i, and $\hat{y}_i$ are the number of data points, target, and predicted value, respectively. This is a simple, basic loss to train a molecular property, yet to match the latent space and design a transfer scheme requires more mathematically advanced techniques.

The Riemannian manifold enjoys isometries induced by diffeomorphisms, which guarantee freedom of coordinate choices on the manifold. Thus, one can find a LF around a point on a manifold. A LF means the coordinate patch can be interpreted as Euclidean around its infinitesimal boundary. Note that the input vector can be embedded on a smooth manifold, and its latent space can also be assumed to be smooth since the ML model should always be smooth at an arbitrary point due to the backpropagation training scheme. Hence, by utilizing the model to map encoder space to a LF, the inner product of vectors satisfies a positive semi-definite property. Considering these aspects, we can fix the latent space to a Riemannian. This whole process is closely described in later sections.

3.2 Matching Coordinate Patches

As we mentioned earlier, we interpret the latent vector from the encoder model to be on a Riemannian manifold. Moreover, we assume each downstream task corresponds to a specific coordinate patch on the manifold with a vast overlapping region in molecular property prediction cases. Since one can always find a mapping relation between different coordinate patches, it is crucial to find the specific analytic form of the mapping function. Yet, it is almost impossible to find an analytic form of the mapping function of an unknown manifold with little numerical information. Hence, we utilize a deep learning model to learn the mapping relation instead of finding one with hands.

The coordinate mapping is induced by a Jacobian on a point. Since a vector enjoys isometry on diffeomorphism on a Riemannian manifold and is written in fundamental representation, the coordinate transformation can be expressed as follows:

$$z'^i \equiv \sum_j \frac{\partial z'^i}{\partial z^j} z^j \tag{2}$$

By introducing a model that mimics the above Jacobian and predicts the transformation matrix for a given vector, one can freely transform between coordinates. However, its inverse may not be stable, and moreover, its derivative may also not be stable and cause instability in the backpropagation process. Resolving the issue is critical for training the model. Therefore, we tweak the problem to not just mimic the Jacobian itself but to predict a transformed vector. Furthermore, we utilize an autoencoder scheme to ensure its inverse behavior is accurate and stable.

$$z' = \text{Transfer}(z), \qquad \hat{z} = \text{Transfer}^{-1}(z') \tag{3}$$

We indeed utilize MSE loss for the autoencoder which consists of transfer and its inverse modules.

$$l_{\text{auto}} = \text{MSE}(z, \hat{z}) \tag{4}$$

Here, we restrict the model by imposing an autoencoder loss to match the overlapping region of coordinates. But this is insufficient; thus, stronger conditions are necessary to match coordinates.

Since we are expecting transformations to occur between coordinate patches, the transfer and inverse modules should have the same input and output dimensions. Moreover, as we are trying to match the underlying geometry of overlapping regions, which are in fact the latent space of the autoencoder, one can realize that we can require another condition for the autoencoder. We consider two distinct autoencoders for source and target coordinates that map coordinate patches to same LF back and forth; the transfer output from both should be equal in order to match the geometry of the overlapping region.

$$z'_s = \text{Transfer}_{s\to LF}(z_s) \qquad \hat{z}_s = \text{Transfer}^{-1}{}_{LF\to s}(z'_s) \tag{5}$$

$$z'_t = \text{Transfer}_{t\to LF}(z_t) \qquad \hat{z}_t = \text{Transfer}^{-1}{}_{LF\to t}(z'_t) \tag{6}$$

Where $\text{Model}_{s\to LF}$ indicate a model from source to LF frame and vice versa.

$$l_{cons} = \text{MSE}(z'_s, z'_t) \tag{7}$$

This loss, we call it a consistency loss, pivots points on overlapping regions to glue the underlying geometry. Furthermore, one can induce information flow from the source domain to the target domain by utilizing a mixture of autodencoders.

$$z'_s = \text{Transfer}_{s\to LF}(z_s) \qquad \hat{z}_{s\to t} = \text{Transfer}^{-1}{}_{LF\to t}(z'_s) \tag{8}$$

The above equation indicates a vector from the source domain that is transformed to the target domain, retaining its mutual information. Therefore, this mapped vector $\hat{z}_{s\to t}$ should also predict the same label as the original vector z_t on the target side. This fact gives rise to another form of loss function, namely a mapping loss.

$$l_{map} = \text{MSE}(y_t, \hat{y}_{s\to t}) \tag{9}$$

Here y_t indicates the label for corresponding vectors and $\hat{y}_{s\to t}$ indicates predicted value from $\hat{z}_{s\to t}$. Above loss ensures mutual information flow by matching overlapping coordinates on pivot points. This particular loss is applied to reverse case too which maps z_t to $\hat{y}_{t\to s}$.

Now the question is, 'Is this enough?'. The answer is 'No!'. The reason is actually not intricate to find. If we start with immensely much data on both the source and target domains, then the answer will be 'Yes'. However, in most cases, we lack data points to glue coordinates together. If points are not enough, there are always overfitting issues that cause the geometry to crumple outside the pivot point region. There are two solutions to this issue. One is gathering more data points and hoping to work smoothly, and the other is by adding a stronger condition to the geometry.

Our idea is not only to focus on point data but also on its geometries by considering the geodesics of two different points. Here, there are two issues to compute the geodesics in curved space: one is that the space itself is curved, and the Pythagorean theorem no longer holds on the space in general.

The other is that the shortest path between two points in a curved space is no longer intuitive. In curved space, one can calculate distance using the following equation:

$$S^2 = \int_l \sum_\mu \sum_\nu g_{\mu\nu} dx^\mu dx^\nu \tag{10}$$

Here, $g_{\mu\nu}$ is a metric on the surface, dx^μ, dx^ν are infinitesimal displacements, and l is the geodesic path between two points. The geodesic of a curved space can be defined as a curve whose parallel transport along the curve preserves the tangent vector to the curve. Which indicates,

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0 \tag{11}$$

Where $\gamma(t)$ is a curve on a manifold, $\dot{\gamma}(t)$ is the derivative respect to a parameter t, and ∇ is a covariant derivative with an affine connection $\Gamma^\mu_{\nu\rho}$. It is clear that solving the geodesic equation is not a simple task since one must know the explicit form of the metric of space, and the equation itself is a non-linear partial differential equation. Thus, as depicted in Figure 2, we introduce infinitesimal perturbation points around a data point to detour the issue. The infinitesimal perturbation makes things easy to tackle since a data point can always be expressed on a LF, where a geodesic becomes trivial. Then one can easily compute the distance between a data point and perturbed points since it can be interpreted to be on a flat Euclidean space.

The distance between two different points on a space is a key quantity to match the task coordinates since, as shown in eq. 10, distance always requires metric on a space. By restricting neighbor points to an infinitesimal region around a pivot data point, the metric can now be considered a Euclidean metric, $\eta_{\mu\nu}$. Than the eq. 10 simplified as follows:

$$S^2 = \int_l \sum_\mu \sum_\nu g_{\mu\nu} dx^\mu dx^\nu = \int_l \sum_\mu \sum_\nu \eta_{\mu\nu} dx^\mu dx^\nu = \int_a^b dx^2 \tag{12}$$

Where a and b are the pivot data point and its perturbation point, respectively. If perturbation is infinitesimal, one can take off the integral and find displacement in a simplified form.

$$S = |b - a| \tag{13}$$

Since it is possible to find a LF on a Riemannian manifold, for a given data point, two different task coordinates can always be transformed by their diffeomorphisms to be on a flat coordinate. If we utilize the above distance between pivot and perturbation points, each task coordinate is now glued together by not only a given pivot point but also by their neighbors. It is closely depicted in Figure 2. Right after the embedding, we find perturbation points around given data points and feed them to each task-specific model for encoding. After encoding, models will be entangled together by mapping loss, consistency loss, and distance loss, forcing the distances between pivot points and their perturbations to be equal between tasks.

$$l_{dis} = \frac{1}{M} \sum_i^M \mathrm{MSE}(s_s^i, s_t^i) \tag{14}$$

Where M is the number of perturbations and s_s^i is the displacement between pivot data points and their perturbations.

$$s_s^i \equiv |(z_s') - (z_s'^i)| \qquad s_t^i \equiv |(z_t') - (z_t'^i)| \tag{15}$$

$$z_s'^i = \mathrm{Transfer}_{s \to LF}(\mathrm{Encoder}_s(x^i)) \tag{16}$$

$$z_t'^i = \mathrm{Transfer}_{t \to LF}(\mathrm{Encoder}_t(x^i)) \tag{17}$$

Here x^i denotes ith perturbation of embedded x, and $\mathrm{Encoder}_s$ and $\mathrm{Encoder}_t$ are encoder parts of source and target model respectively. Finally, by gathering all losses with individual hyperparameters, we obtain the complete form of the loss function used in the *GATE* algorithm.

$$l_{tot} = l_{reg} + \alpha l_{auto} + \beta l_{cons} + \gamma l_{map} + \delta l_{dis} \tag{18}$$

Each hyperparameter tunes the corresponding weight of loss. In most cases, it is sufficient to let all the hyperparameters to 1, but in some cases, one can fine-tune the hyperparameters to achieve superior performance. The most critical hyperparameter is δ since the distance loss manages the regularization effect of the model. This result will be closely examined in the ablation section. And with the proper hyperparameters, our novel transfer model *GATE* outperforms conventional models in inductive transfer settings for regression tasks, as shown in the experiment section.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

To test our algorithms, we used a total of 14 datasets from three different open databases named PubChem(Kim et al., 2022), Ochem(Sushko et al., 2011), and CCCB(III, 2022), as described in Appendix Table 3. The training and test split is an 80:20 ratio, and two sets of data were prepared: random split and scaffold-based split(Bemis & Murcko, 1996). Every experiment is tested in a four-fold cross-validation setting with uniform sampling for accurate evaluation, and a single NVIDIA A40 is used for the experiments. For the evaluation, we compare the performance of *GATE* against that of single task learning (STL), MTL, KD, global structure preserving loss based KD (GSP-KD) (Joshi et al., 2022), and transfer learning (retrain all or head network only). We used the same architecture for encoders and heads in both the baselines and our model for all experiments. More detailed experimental setups can be found in Appendix. We performed experiments with a total of 23 target and source task pairs, respectively.

4.2 RESULTS

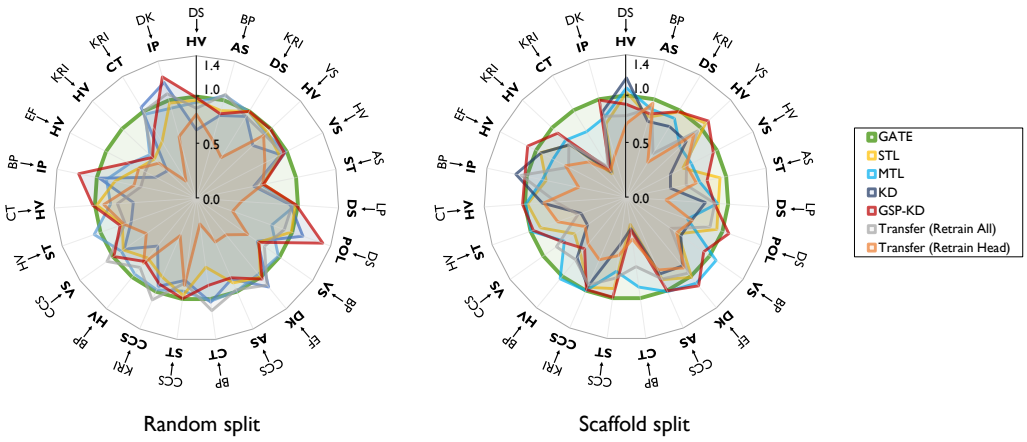

Figure 3: The RMSE of *GATE* divided by the RMSE of the corresponding methods are illustrated in the radar chart. Experiments were performed with 23 target and source task pairs in random split (left) and scaffold split (right) datasets. Inner characters correspond to the target tasks, and outer characters are for the source tasks. The full names of the tasks can be found in Appendix Table 3.

The result of the training for the 23 task pairs with both random and scaffold split is demonstrated in Figure 3. The figure illustrates the relative regression accuracy, calculated as the RMSE of *GATE* divided by the RMSE of the corresponding method. As shown in the figure, in the majority of cases, *GATE* outperforms baseline methods by significant margins. Note that, even though the KD and GSP-KD outperforms the *GATE* for some task pairs, they show critically low accuracy for several task pairs. On the other hand, as we can see in the Appendix Table 4-7, *GATE* exhibited the best performance in more than half of task pairs, specifically 12 pairs in random split and 13 pairs in scaffold split out of 23 pairs. Considering that the method with the second-highest number of best-performing task pairs had only 4 task pairs in random split and 5 task pairs in scaffold split, this can be regarded as a dramatic difference. Also, when considering the second-best performance, *GATE* maintained consistently high performance in the majority of pairs, reaching 18 pairs in random split and scaffold split. For the average RMSE over all task pairs, *GATE* demonstrates significantly lower RMSE compared to the second-best method, 9.8 % lower in random split (GSP-KD) and 14.3 % lower in scaffold split (MTL). This indicates that *GATE* performs even better in extrapolation tasks.

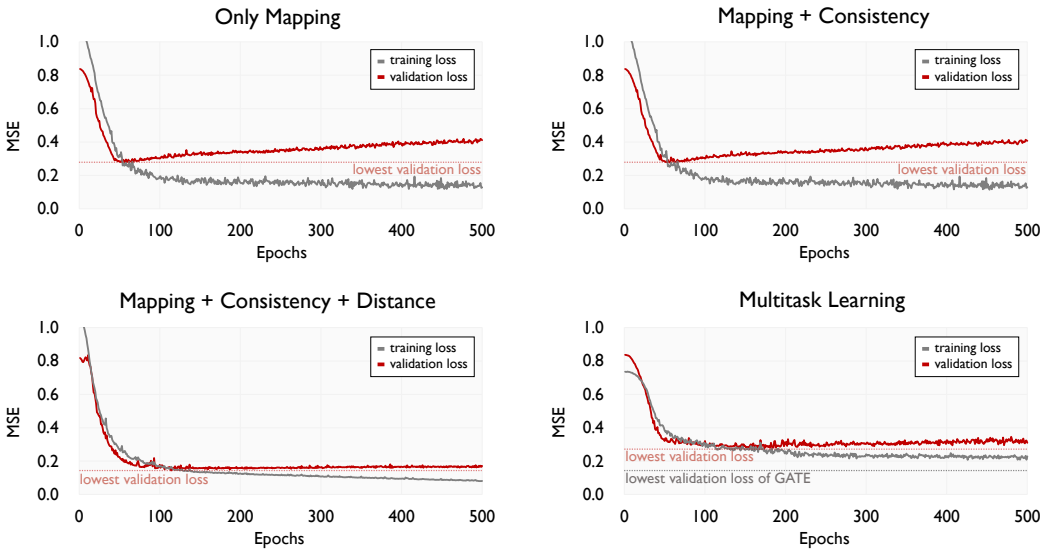

Figure 4: Training and validation loss curves of *GATE* and MTL models with different sets of loss terms. Upper left: *GATE* trained without consistency and distance loss; upper right: *GATE* trained without distance loss; lower left: plain *GATE*, lower right: plain MTL.

5 Ablations Studies and Further Analysis

5.1 Role of Distance Loss

The most distinguishable aspect of *GATE* is the distance loss, which connects task coordinates not only through labeled data points but also surrounding perturbation points. Since learning signals from consistency and distance loss do not require regression target labels, these two losses can be applied in an unsupervised manner. *GATE* directly leverages this advantage in distance loss and regularizes the overall latent space to be geometrically transferable to a LF. To verify this, we trained *GATE* using different sets of losses: 1) only mapping, 2) mapping and consistency, and 3) mapping, consistency, and distance losses. As a result, we observed that the addition of distance loss significantly suppresses overfitting during the training process, as shown in Figure 4. Note that the training loss curve of MTL never reached the lowest validation loss of *GATE* (grey dashed line). These results indicate that the regression model can be strongly regularized by the distance loss while having a minimal adverse impact on the regression task itself.

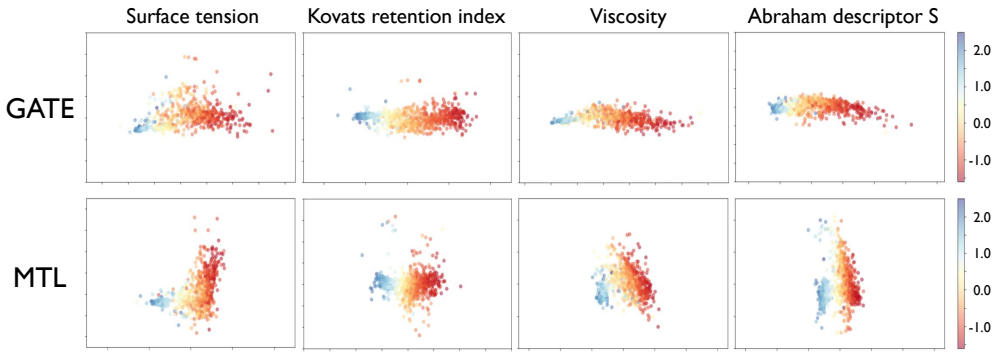

Figure 5: PCA on latent space of *GATE* and MTL model trained with collision cross section dataset as a target task and surface tension, Kovats retention index, viscosity, and Abraham descriptor S as source tasks, respectively. The color indicates the predicted value of the collision cross section.

5.2 STABILITY OF LATENT SPACE

For further investigation on the regularization effect of *GATE*, we examined the shape of the latent space of the *GATE* and MTL models. Transfer learning offers a way to leverage information from source tasks, but undesired interfering information can cause negative transfer. Ideally, if a model is well-guided by the right information and regularized properly, the overall geometry of the latent space may remain stable and not depend on the type of source tasks. (Zhang et al., 2023; Gómez-Bombarelli et al., 2018) However, if the target task is overwhelmed by the source task and regularization is not enough, latent space will be deformed according to the source tasks. To check this effect, we trained regression on the same target task using *GATE* or MTL with four different source tasks and then performed principal component analysis. Figure 5 effectively illustrates that the latent space of *GATE* exhibits relatively stable characteristics compared to that of MTL.

5.3 ROBUSTNESS TO DATA CORRUPTION

We examined the robustness against data corruption in order to directly verify the regularization effect. To artificially introduce significant corruption to the dataset, data points with labeled values outside the standard deviation were randomly selected, and the values of these data points were altered to be twice the standard deviation with negation. And then, after training with the corrupted dataset, we assessed the mean squared error (MSE) between the model's predictions on the corrupted data points used during training and the original values before the corruption. If the model exhibits overfitting to the corrupted data, it will manifest substantial errors, whereas effective model regularization will lead to error reduction. Figure 6 shows the result of the test for *GATE* and MTL. The *GATE* model exhibits considerably lower errors compared to MTL. This result indicates that the regularization effect of the *GATE* algorithm leads to high robustness against data corruption and successful generalization.

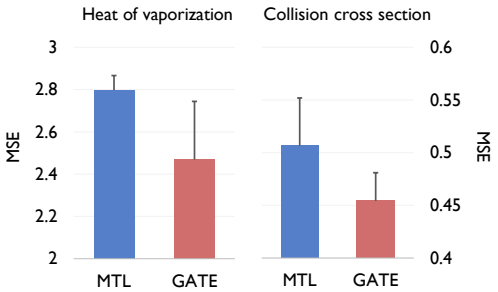

Figure 6: *GATE* and MTL models are trained with artificially corrupted data points and MSEs between predicted values and original values before the corruption is recorded. The target task is heat of vaporization (left) or collision cross-section (right), with the opposite side being used as the source task.

6 DISCUSSION

We demonstrated a novel transfer algorithm based on the geometrical interpretation of latent space. By assuming latent space to be on a Riemannian manifold with a curved metric, the conventional Euclidean interpretation is no longer valid. However, since diffeomorphism invariance guarantees freedom of coordinate choices, one can always find a LF for each task coordinate. Hence, it is possible to impose consistency loss by matching an encoded vector from one task to another. Furthermore, we showed that the distance loss of displacement from a point in an overlapping region to its perturbations is crucial. All together, we demonstrated that the *GATE* outperforms conventional methods in the majority of molecular prediction task setups and proved its robustness.

In this article, we focused on regression tasks for molecular property prediction. Yet, the algorithm itself is not restricted to regression tasks. Thus, it is interesting to expand the algorithm to other domains such as images, languages, or, further, multi-modal setups. Another technical point is that we considered the neighboring region of a pivoting point to compute displacement on a flat space. However, to match the entire geometry, it is necessary to consider not only perturbations but also points in finite-distance regions. In this scheme, one must solve a geodesic equation instead of merely computing the Euclidean distance. It is incomparably complicated since one must consider 1) finding an explicit form of curved metric and 2) solving highly non-linear differential geodesic equations analytically or numerically. Yet, considering *GATE*'s performance, by imposing geometric characters on fully curved backgrounds, the performance gains may be substantial. Hence, this could be another interesting research topic to expand *GATE*'s horizons.

REFERENCES

Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Domain adaptation on the statistical manifold. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2481–2488, 2014.

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

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. 12 2013.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In *ICML*, pp. 1749–1759, 2021.

Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10:370–377, 2019. doi: 10.1039/C8SC04228D. URL http://dx.doi.org/10.1039/C8SC04228D.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 06 2016.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems (NIPS)*, 13, 09 2015.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 2066–2073. IEEE, 2012.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pp. 999–1006. IEEE, 2011.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Noah Hoffmann, Jonathan Schmidt, Silvana Botti, and Miguel AL Marques. Transfer learning on large datasets for the accurate prediction of material properties. *arXiv preprint arXiv:2303.03000*, 2023.

Russell D. Johnson III. Nist computational chemistry comparison and benchmark database. *NIST Standard Reference Database*, 101, 2022. URL http://cccbdb.nist.gov/.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization, 12 2018.

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.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 10 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac956. URL https://doi.org/10.1093/nar/gkac956.

Sung Moon Ko, Sungjun Cho, Dae-Woong Jeong, Sehui Han, Moontae Lee, and Honglak Lee. Grouping matrix based graph pooling with adaptive number of clusters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8334–8342, 2023.

Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. *CVPR 2011*, pp. 1785–1792, 2011. URL https://api.semanticscholar.org/CorpusID:7419723.

Kyoungyeul Lee and Dongsup Kim. In-silico molecular binding prediction for human drug targets using deep neural multi-task learning. *Genes*, 10(11):906, 2019.

Xinhao Li and Denis Fourches. Inductive transfer learning for molecular activity prediction: Next-gen qsar models with molpmofit. *Journal of Cheminformatics*, 12(1):1–15, 2020.

Shengchao Liu, Meng Qu, Zuobai Zhang, Huiyu Cai, and Jian Tang. Structured multi-task learning for molecular property prediction. In *International conference on artificial intelligence and statistics*, pp. 8906–8920. PMLR, 2022.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Wei Cheng, Xiang Zhang, and Wei Wang. *Dual Transfer Learning*, pp. 540–551. doi: 10.1137/1.9781611972825.47. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972825.47.

You-Wei Luo, Chuan-Xian Ren, Pengfei Ge, Ke-Kun Huang, and Yu-Feng Yu. Unsupervised domain adaptation via discriminative manifold embedding and alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5029–5036, 2020.

Jianhan Pan, Teng Cui, Thuc Duy Le, Xiaomei Li, and Jing Zhang. Multi-group transfer learning on multiple latent spaces for text classification. *IEEE Access*, 8:64120–64130, 2020. doi: 10.1109/ACCESS.2020.2984571.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Minshi Peng, Yue Li, Brie Wamsley, Yuting Wei, and Kathryn Roeder. Integration and transfer learning of single-cell transcriptomes via cfit. *Proceedings of the National Academy of Sciences*, 118(10):e2024383118, 2021. doi: 10.1073/pnas.2024383118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2024383118.

Ariadna Quattoni, Michael Collins, and Trevor Darrell. Transfer learning for image classification with sparse prototype representations. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 03 2008. doi: 10.1109/CVPR.2008.4587637.

Maithra Raghu, Chiyuan Zhang, Jon M. Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning with applications to medical imaging. *CoRR*, abs/1902.07208, 2019. URL http://arxiv.org/abs/1902.07208.

Franco Scarselli, Marco Gori, Ah Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 20:61–80, 01 2009. doi: 10.1109/TNN.2008.2005605.

Iurii Sushko, Sergii Novotarskyi, Robert Körner, Anil Kumar Pandey, Matthias Rupp, Wolfram Teetz, Stefan Brandmaier, Ahmed Abdelaziz, Volodymyr V Prokopenko, Vsevolod Y Tanchuk, et al. Online chemical modeling environment (ochem): web platform for data storage, model development and publishing of chemical information. *Journal of computer-aided molecular design*, 25:533–554, 2011.

Jingshu Wang, Divyansh Agarwal, Mo Huang, Gang Hu, Zilu Zhou, Chengzhong Ye, and Nancy Zhang. Data denoising with transfer learning in single-cell transcriptomics. *Nature Methods*, 16: 875–878, 09 2019. doi: 10.1038/s41592-019-0537-1.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

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.

Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge University Press, 2020.

Xiang Yu, Jian Wang, Qing-Qi Hong, Raja Teku, Shui-Hua Wang, and Yu-Dong Zhang. Transfer learning for medical images analyses: A survey. *Neurocomputing*, 489:230–254, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.08.159. URL https://www.sciencedirect.com/science/article/pii/S0925231222003174.

Liang Zeng, Lanqing Li, and Jian Li. Molkd: Distilling cross-modal knowledge in chemical reactions for molecular property prediction. *arXiv preprint arXiv:2305.01912*, 2023.

Hao Zhang, Nan Zhang, Ruixin Zhang, Lei Shen, Yingyi Zhang, and Meng Liu. Coordinating cross-modal distillation for molecular property prediction. *arXiv preprint arXiv:2211.16712*, 2022.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2023. doi: 10.1109/JAS.2022.106004.

Fuzhen Zhuang, Ping Luo, Hui Xiong, Qing He, Yuhong Xiong, and Zhongzhi Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization†. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):100–114, 2011. doi: https://doi.org/10.1002/sam.10099. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.10099.

Fuzhen Zhuang, Ping Luo, Changying Du, Qing He, and Zhongzhi Shi. Triplex transfer learning: Exploiting both shared and distinct concepts for text classification. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pp. 425–434, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318693. doi: 10.1145/2433396.2433449. URL https://doi.org/10.1145/2433396.2433449.

Fuzhen Zhuang, Ping Luo, Changying Du, Qing He, Zhongzhi Shi, and Hui Xiong. Triplex transfer learning: Exploiting both shared and distinct concepts for text classification. *IEEE Transactions on Cybernetics*, 44(7):1191–1203, 2014. doi: 10.1109/TCYB.2013.2281451.