Generative Adversarial Symmetry Discovery

Jianke Yang [1] **Robin Walters** [*2] **Nima Dehmamy** [*3] **Rose Yu** [1]

Abstract

Despite the success of equivariant neural networks in scientific applications, they require knowing the symmetry group a priori. However, it may be difficult to know which symmetry to use as an inductive bias in practice. Enforcing the wrong symmetry could even hurt the performance. In this paper, we propose a framework, LieGAN, to *automatically discover equivariances* from a dataset using a paradigm akin to generative adversarial training. Specifically, a generator learns a group of transformations applied to the data, which preserve the original distribution and fool the discriminator. LieGAN represents symmetry as interpretable Lie algebra basis and can discover various symmetries such as the rotation group $\mathrm{SO}(n)$, restricted Lorentz group $\mathrm{SO}(1,3)^+$ in trajectory prediction and top-quark tagging tasks. The learned symmetry can also be readily used in several existing equivariant neural networks to improve accuracy and generalization in prediction. Our code is available at https://github.com/Rose-STL-Lab/LieGAN.

1. Introduction

Symmetry is an important inductive bias in deep learning. For example, convolutional neural networks (Krizhevsky et al., 2017) exploit translational symmetry in images, and graph neural networks utilize permutation symmetry in graph-structured data (Kipf & Welling, 2016). Equivariant networks have led to significant improvement in generalization, sample efficiency and scientific validity (Zaheer et al., 2017; Weiler & Cesa, 2019; Cohen et al., 2019a; Wang et al., 2021). Interest has surged in both theoretical analysis and practical techniques for building general group equivariant neural networks (Kondor & Trivedi, 2018; Cohen et al., 2019b; Bekkers, 2019; Finzi et al., 2021).

[*]Equal contribution [1]University of California San Diego [2]Northeastern University [3]IBM Research. Correspondence to: Rose Yu <roseyu@ucsd.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

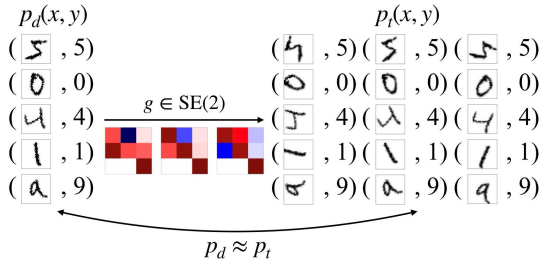


Figure 1. Connection between symmetry and data distribution. MNIST classification is invariant to a subset of $\mathrm{SE}(2)$ transformations. If the digits are transformed by random rotations and translations within this subset, the resulting data distribution p_t remains close to the original distribution p_d.

However, a key limitation of equivariant neural networks is that they require explicit knowledge of the data symmetry before a model can be constructed. In practice, it is sometimes difficult to identify the true symmetries of the data, and constraining the model by the exact mathematical symmetry might not be optimal in real-world situations (Wang et al., 2022). These challenges call for approaches that can automatically discover the underlying symmetry of the data.

Neural networks that discover unknown symmetry play the role of AI scientists , not only by making data-driven predictions, but also by identifying and describing physical systems through their symmetries and generating new scientific insights through the close relationship between symmetry, conservation laws and underlying governing equations (Alet et al., 2021). Most existing work on symmetry discovery can only address a small fraction of potential symmetry types, such as finite groups (Zhou et al., 2020), subsets of a given group (Benton et al., 2020) or individual group elements (Desai et al., 2021). L-conv (Dehmamy et al., 2021) can discover continuous symmetries without discretization of the groups, but has limited computational efficiency. A more general framework is needed for the discovery of various real-world symmetries.

In this work, we present a novel framework for discovering continuous symmetry from data using generative adversarial training (Goodfellow et al., 2014). We establish the connection between symmetry and data distribution as in figure 1. Our method then trains a symmetry generator that transforms the training data and outputs a similar distribution to the original dataset, which suggests equivariance or invari-

ance to the learned tranformations. Making use of the theory of Lie groups and Lie algebras, our method, *LieGAN*, is able to discover continuous symmetries as matrix groups. Moreover, through different parameterization strategies, it can also deal with other types of symmetries, such as discrete group transformation, as well as the subset of a group.

Our main contributions can be summarized as follows:

1. We propose LieGAN, a method for automatically discovering symmetries from data, capable of learning general linear symmetries, including the rotation group $\mathrm{SO}(n)$ and restricted Lorentz group $\mathrm{SO}(1,3)^+$.
2. LieGAN is interpretable, directly yielding an orthogonal Lie algebra basis as a discovery result.
3. We show that the Lie algebra learned by LieGAN leads to excellent performance in downstream tasks such as N-body dynamics and top quark tagging.
4. We propose LieGNN, a modified E(n) Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021) that integrates symmetries learned by LieGAN, achieving similar performance to equivariant models with ground truth symmetries.

2. Related Work

Equivariant Neural Networks. Many works have addressed the problems of designing neural network modules that are equivariant to specific transformations, such as permutation in sets (Zaheer et al., 2017), local gauge transformations (Cohen et al., 2019a), scaling (Worrall & Welling, 2019), rotation on spheres (Cohen et al., 2018) and general $\mathrm{E}(2)$ transformations on Euclidean plane (Weiler & Cesa, 2019). Another branch of works focus on developing theoretical guidelines and practical methods for building general group equivariant neural networks (Cohen & Welling, 2016; Kondor & Trivedi, 2018; Cohen et al., 2019b; Finzi et al., 2020; 2021). However, these methods rely on explicit a priori knowledge of the data symmetry. Instead, we are interested in discovering knowledge of the symmetry itself. The learned symmetry can then be used to select or design an equivariant neural network to make predictions.

Generative Adversarial Training. The original generative adversarial network (GAN) (Goodfellow et al., 2014) uses a generator to transform random noise into target distribution. Many variants of GAN have been proposed to address different tasks other than the unrestricted generation (Mirza & Osindero, 2014; Karras et al., 2019; Isola et al., 2016; Zhu et al., 2017; Antoniou et al., 2017). In particular, CycleGAN (Zhu et al., 2017) learns a generator that maps the input image to another domain. DAGAN (Antoniou et al., 2017) also takes data points from a source domain and generalizes them to a broader domain with a generator to perform data augmentation, which is related to our task, as the augmenting process can be regarded as a set of transformations to which the data is invariant. These works use samples from the original distribution instead of random noise as generator input and perform domain transfer or generalization with a generator. Our work proposes another usage of such design. The generator in our model produces transformations that are applied to data samples, and discovers the underlying symmetry by learning the correct set of transformations.

Symmetry Discovery. Many existing symmetry discovery methods (Benton et al., 2020; Zhou et al., 2020; Romero & Lohit, 2021; Krippendorf & Syvaeri, 2020) limit their search space to a small fraction of potential symmetry types. MSR (Zhou et al., 2020) reparameterizes network weights into task weights and a symmetry matrix and meta-learns the symmetry matrix to provide information about task symmetry. However, it can only be applied to finite groups and scales linearly in space complexity with the size of the group, which eliminates the possibility of applying this algorithm to infinite continuous groups. Augerino (Benton et al., 2020) addresses a different but relevant scenario: learning the extent of symmetry within a given group. Partial G-CNN (Romero & Lohit, 2021) also learns group subsets via distributions on group to describe the symmetry at different levels in the model. These approaches can only be applied to cases where the symmetry group is known. Krippendorf & Syvaeri (2020) proposes to detect symmetries by constructing a synthetic classification task and examining the structure of network embedding layers. This method involves some manual procedures, such as defining the classification task and choosing the metric for latent space analysis. Our work aims to address all of the above limitations within a unified, automated framework.

L-conv (Dehmamy et al., 2021) develops a Lie algebra convolutional network that can model any group equivariant functions. However, it performs first order approximation for matrix exponential and uses recursive layers to push the kernel away from the identity, which may become too expensive in practice. Moreover, the experiments are limited to image datasets.

Desai et al. (2021) also proposes to discover symmetries of the dataset distribution with a GAN. A major limitation of their algorithm is that the model can only learn one group element in a single round of training and has to rely on other techniques such as subgroup regularization or group composition to identify the group. Also, their definition of symmetry is different from ours.

Comparison between LieGAN and other works on symmetry discovery can be found in Table 1. To the best of our

knowledge, our approach is the *first* to address the discovery of such a variety of symmetries including discrete group, continuous group, and subset of given or unknown group.

Table 1. Comparison of different models' capability of discovering different kinds of symmetries

SYMMETRY	MSR	AUGERINO	LIEGAN
DISCRETE	✓	✗	✓
CONTINUOUS	✗	✗	✓
GIVEN GROUP SUBSET	✗	✓	✓
UNKNOWN GROUP SUBSET	✗	✗	✓

3. Background

Before presenting our methodology, we provide some preliminary concepts that appear frequently in our work. We assume basic knowledge about group theory.

Lie group. A Lie group is a group that is also a differentiable manifold. It can be used to describe continuous transformations. For example, all 2D rotations form a Lie group $\mathrm{SO}(2)$, where rotation with angle θ can be represented by $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. Also, all Euclidean transformations including reflection, rotation and translation form the Lie group of $\mathrm{E}(n)$. Each Lie group is associated with a Lie algebra, which is its tangent vector space at identity: $\mathfrak{g} = T_{\mathrm{Id}}G$. The basis of the Lie algebra $L_i \in \mathfrak{g}$ are called (infinitesimal) generators of the Lie group. Group elements that are infinitesimally close to identity can be written in terms of these generators: $g = \mathrm{Id} + \sum_i \epsilon_i L_i$.

We can use an exponential map $\exp : \mathfrak{g} \to G$ to map Lie algebra elements to Lie group elements. For matrix groups, matrix exponential is such a map. For a connected Lie group G, its elements can be written as $g = \exp(\sum_i w_i L_i)$.

Group representation. We are interested in how group elements transform the data. We assume that the input space is $\mathcal{X} = \mathbb{R}^n$. A group element $g \in G$ can act linearly on $x \in \mathcal{X}$ via $\rho_{\mathcal{X}}(g)$, where $\rho_{\mathcal{X}} : G \to \mathrm{GL}(n)$ is a group representation. $\rho_{\mathcal{X}}$ maps each group element g to a nonsingular matrix $\rho_{\mathcal{X}}(g) \in \mathbb{R}^{n\times n}$ that transforms the input vector.

A group representation $\rho : G \to \mathrm{GL}(n)$ induces a representation for the Lie algebra $\mathfrak{g} = T_{\mathrm{Id}}G$ denoted as $d\rho : \mathfrak{g} \to \mathfrak{gl}(n)$, which relate to the representation of its Lie group by $\exp(d\rho(L)) = \rho(\exp(L))$.

4. Symmetry Discovery

We aim to automatically discover symmetry from data. Formally, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a dataset with distribution $x_i, y_i \sim p_d(x, y)$, input space $\mathcal{X} = \mathbb{R}^n$, output space $\mathcal{Y} = \mathbb{R}^m$ and an unknown function $f : \mathcal{X} \to \mathcal{Y}$. We have:

Definition 1 (Equivariance)**.** Suppose a group G acts on $\mathcal{X}$ and $\mathcal{Y}$ via representations $\rho_{\mathcal{X}} : G \to \mathrm{GL}(n)$ and $\rho_{\mathcal{Y}} : G \to \mathrm{GL}(m)$. Then, a function $f : \mathcal{X} \to \mathcal{Y}$ is *equivariant* if $\forall g \in G$, $(x, y) \in \mathcal{D}$, $\rho_{\mathcal{Y}}(g)y = f(\rho_{\mathcal{X}}(g)x)$. We omit $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ when clear and write $gy = f(gx)$.

We also address invariance in this work, which is a special case of equivariance when $\rho_{\mathcal{Y}}(g) = \mathrm{Id}$. Next, we describe the formulation to relate symmetry discovery with generative adversarial training (Goodfellow et al., 2014).

4.1. Generative Adversarial Symmetry Discovery

By Definition 1, if a group element acts on the input, the output of an equivariant function is also transformed correspondingly by the representation of the same element. From another perspective, if all the data samples are transformed in this way, the transformed data distribution should remain similar to the original dataset distribution, as is demonstrated in figure 1.

At a high level, we want to design a generator that can efficiently produce transformed input and a discriminator that can not distinguish real samples from the dataset and the outputs from the generator. Through adversarial training, the generator tries to fool the discriminator by learning a group of transformations that minimize the divergence between the transformed and the original distributions. This group of transformations defines the symmetry of interest.

We present our symmetry discovery framework in figure 2. We are interested in how the group G acts on data through its representations $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$. We learn G as a subgroup of $\mathrm{GL}(k)$ for some k chosen based on the task. The representations $\rho_{\mathcal{X}} \colon \mathrm{GL}(k) \to \mathrm{GL}(n)$ and $\rho_{\mathcal{Y}} \colon \mathrm{GL}(k) \to \mathrm{GL}(m)$ are also chosen and fixed based on the task. The GAN generator Φ samples an element from a distribution μ defined on $\mathrm{GL}(k)$ and then applies it to x and y:

$$\Phi(x, y) = (\rho_{\mathcal{X}}(g)x, \rho_{\mathcal{Y}}(g)y) \tag{1}$$

For an invariant task, for example, we set $k = n$ and $\rho_{\mathcal{X}} = \mathrm{Id}$ the standard representation and $\rho_{\mathcal{Y}} = 1$ the trivial representation. For a time series prediction task, predicting a system state based on t previous states, we set $k = m$ and $n = tm$ and $\rho_{\mathcal{X}} = \mathrm{Id}^{\oplus t}$ and $\rho_{\mathcal{Y}} = \mathrm{Id}$.

Under this formulation, the generator should learn a subgroup of the transformations to which $f : \mathcal{X} \to \mathcal{Y}$ is equivariant. That is, it should generate a distribution close to the original data distribution. Similar to the setting in GAN, we optimize the following minimax objective:

$$\begin{aligned}&\min_{\Phi}\max_{D} L(\Phi, D)\\ =&\mathbb{E}_{x,y\sim p_d, g\sim\mu}\Big[\log D(x, y) + \log(1 - D(\Phi(x, y)))\Big]\end{aligned}$$

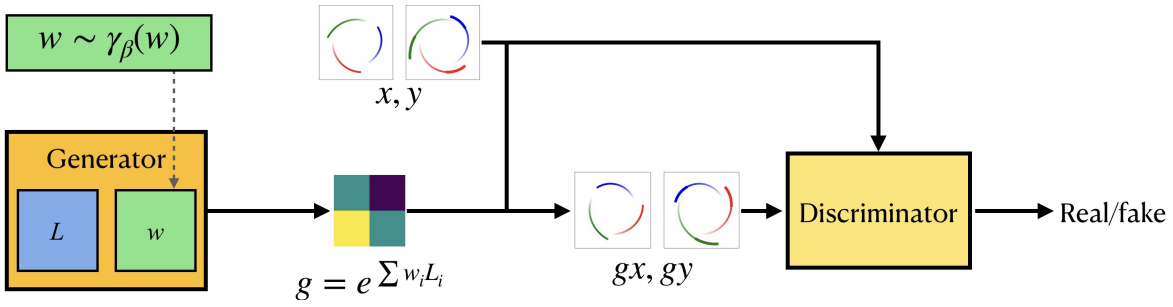


Figure 2. Structure of the proposed LieGAN model. The transformation generator learns a continuous Lie group acting on the data that preserves the original joint distribution. For example, this figure shows a task of predicting future 3-body movement based on past observations, where the generator could learn rotation symmetry.

$$=\mathbb{E}_{x,y\sim p_d}[\log D(x,y)] + \mathbb{E}_{x,y\sim p_g}[\log(1 - D(x,y))] \quad (2)$$

where D is a standard GAN discriminator that outputs a real value as the probability that (x, y) is a real sample, p_d is the density of the original data distribution and p_g is the generator-transformed distribution given according to change-of-variable formula by

$$p_g(x,y) = \int_g \mu(g) p_d(g^{-1}x, g^{-1}y)/(|\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)|)dg \quad (3)$$

Under the ideal discriminator, the generator in the original GAN formulation minimizes the JS divergence between two distributions (Nowozin et al., 2016). In our setting, we prove that our generator can achieve zero divergences with the correct symmetry group under certain circumstances.

Theorem 1. *The generator can achieve zero JS divergence by learning a maximal subgroup $G^* \subset \mathrm{GL}(n)$ with respect to which $y = f(x)$ is equivariant if $p_d(x)$ is distributed proportionally to the volume of inverse group element transformation along each orbit of G^*-action on $\mathcal{X}$, that is, $p_d(gx_0) \propto |\rho_{\mathcal{X}}(g^{-1})||\rho_{\mathcal{Y}}(g^{-1})|$.*

The hypothesis of Theorem 1 is equivalent to saying that $p_d(x)$ is uniform along each group action orbit when the transformation is volume preserving, as in the case of rotation. However, as this is often not satisfied in practice, there is no guarantee that the generator can achieve zero divergences with nonidentical transformations. Nevertheless, as formalized in the following theorem, the generator can learn a nontrivial symmetry under some weak assumptions.

Theorem 2. *Under assumptions 1, 2 and 3, the GAN loss function under the ideal discriminator $L(\Phi, D^*)$ is lower with a generator that learns a subspace of the true Lie algebra $\mathfrak{g}^*$ than a generator with an orthogonal Lie algebra to $\mathfrak{g}^*$. That is, if $\mathfrak{g}_1 \cap \mathfrak{g}^* \neq \{\mathbf{0}\}$, $\mathfrak{g}_2 \cap \mathfrak{g}^* = \{\mathbf{0}\}$, then $L(\mathfrak{g}_1, D^*) < L(\mathfrak{g}_2, D^*) = 0$.*

Theorem 2 ensures that a partially correct symmetry results in lower loss function value than an incorrect symmetry. In other words, optimizing (2) leads to symmetry discovery.

The related assumptions and proofs for Theorem 1 and 2 are deferred to Appendix A.1.

4.2. Parameterizing Distributions Over Lie Group

We use the theory of Lie groups to model continuous sets of transformations. To parameterize a distribution on a Lie group with c dimensions and k representation dimensions, our model learns Lie algebra generators $\{L_i \in \mathbb{R}^{k\times k}\}_{i=1}^{c}$ and samples the coefficients $w_i \in \mathbb{R}$ for their linear combination from either a fixed or a learnable distribution. The Lie algebra element is then mapped to a Lie group element using the matrix exponential (Falorsi et al., 2019).

$$w \sim \gamma_\beta(w), \quad g = \exp\Big[\sum_i w_i L_i\Big] \quad (4)$$

The coefficient distribution γ_β can be either fixed or updated, depending on our focus of discovery. If we have little information on the group, then by learning the L_i and leaving the coefficient distribution fixed, our model can still express distributions over many different groups. On the other hand, we may want to find a subgroup or a subset of some known group. For example, the symmetry may be some discrete subgroup of SO(2) for some tasks. In this case, we fix L as the rotation generator and learn γ_β, revealing peaks at certain values. Learning γ_β is also useful when the task is not equivariant to the full group, but displays invariance for a subset of transformations, for example, the case of MNIST image classification, where the rotation by π will obscure the boundary between "6" and "9" (Benton et al., 2020). Generally, allowing for β to be learnable gives the model more freedom to discover various symmetries.

The coefficient distribution γ_β may be parameterized in different ways. A normal distribution centered at the origin is a natural choice, since it assigns the same probability density for a group element and its inverse, and the variance can either be fixed or learned through the reparameterization

trick (Kingma & Welling, 2013). However, a multimodal distribution like a Gaussian mixture model may be better at capturing discrete subgroups.

However, we note that a limitation of using the Lie algebra to parameterize transformations is that it can only capture a single connected component of the Lie group. Some groups such as $\mathrm{E}(n)$ do not have a surjective exponential map and their group elements must be described by introducing additional discrete generators: $g = \exp[\sum_i t_i L_i] \prod_j h_j$.

4.3. Regularization Against Trivial Solutions

In our optimization problem (2), the generator can learn a trivial symmetry of identical transformation. We alleviate this issue by penalizing the similarity between the input and output of the generator. Let R be similarity function on $\mathcal{X} \times \mathcal{Y}$. The regularizer is then defined

$$l_{\mathrm{reg}}(x, y) = R(\Phi(x, y), (x, y)). \tag{5}$$

We note that the similarity function need to recognize the difference in data before and after the transformation. In practice, we use cosine similarity, which only has scaling invariance in all dimensions.

Another issue arises when dealing with multi-dimensional Lie groups. The model is encouraged to search through different directions in the manifold of the general linear group with multiple channels, i.e. L_i's. In practice, however, we find that they tend to learn similar elements. To address this problem, we introduce another regularization against the channel-wise similarity, denoted as

$$l_{\mathrm{chreg}}(\Phi) = \sum_{1 \le i < j \le c} R_{\mathrm{ch}}(L_i, L_j) \tag{6}$$

where c is the number of channels in generator and R_{ch} is the cosine similarity for L_i weights. We also consider setting R_{ch} to be the Killing form (Knapp & Knapp, 1996) , a metric defined in the Lie algebra. In this case, minimizing l_{chreg} corresponds to discovering an orthogonal basis for the Lie algebra. In practice, we find that cosine similarity works best.

Combining these regularizers with (2), we optimize the following objective:

$$\begin{aligned} L_{\mathrm{reg}}(\Phi, D) =& \mathbb{E}_{(x,y),g}[\log D(x, y) + \log(1 - D(\Phi(x, y))) \\ &+ \lambda \cdot l_{\mathrm{reg}}(x, y)] + \eta \cdot l_{\mathrm{chreg}}(\Phi) \end{aligned} \tag{7}$$

4.4. Model Architecture

LieGAN consists of two components, the generator and the discriminator. The generator simply samples a Lie group element to transform the input data and does not have any deep neural network. The discriminator can be any network architecture that fits the input. In practice, we use Multilayer Perceptron (MLP) as a discriminator unless otherwise stated. We find that the generator loss is usually higher than the discriminator loss during training, which suggests that the generator's task of finding the correct set of symmetry transformations is harder. Therefore, a simple discriminator architecture is sufficient.

5. Using the Discovered Symmetry

The discovered symmetry can be used as an inductive bias to aid prediction. For instance, Augerino (Benton et al., 2020) develops an end-to-end pipeline that simultaneously discovers invariance and trains an invariant model. For our method, there are multiple ways of utilizing the learned Lie algebra representation in downstream prediction tasks.

5.1. Data Augmentation

A natural idea would be augmenting the training data with the transformation generator in LieGAN, which would lead to better generalization and robustness similar to other data augmentation approaches (Dao et al., 2019). To perform data augmentation in the equivariance scenario, we transform the input with group element g and transform the model output with g^{-1} to obtain the final prediction

$$\hat{y} = g^{-1} f_{\mathrm{model}}(gx). \tag{8}$$

5.2. Equivariant Model

The discovered symmetry from LieGAN can also be easily incorporated into existing equivariant models due to its explicit representation of the Lie algebra. This procedure is specific to different equivariant model architectures. We provide two examples of incorporating the learned symmetry into EMLP (Finzi et al., 2021) and EGNN (Satorras et al., 2021), which are also used in experiments.

EMLP. Finzi et al. (2021) provide a simple interface for building equivariant MLPs for arbitrary matrix groups. We can directly use the discovered Lie algebra basis as input to the method and obtain an MLP equivariant to the corresponding connected Lie group, with only a minor modification to the model. The original EMLP constructs a constraint matrix according to the specified group and projects network weights to its null space. This does not work well with the symmetry discovered by LieGAN, because it inevitably has some numerical error that results in a higher rank constraint matrix and thus lower rank weight matrix. In practice, we raise the singular value threshold to obtain an approximate equivariant subspace with more dimensions. Implementation details can be found in Appendix B.1.

EGNN. Satorras et al. (2021) encode the $\mathrm{E}(n)$ equivariance in a graph neural network (GNN) by computing invariant edge features using the Euclidean metric. Similarly, Gong et al. (2022) develop a Lorentz invariant GNN for jet tagging by computing invariant edge features using the Minkowski metric. Both methods may be summarized as:

$$\begin{aligned} m_{ij} =& \phi_e(h_i, h_j, \|x_i - x_j\|_J^2, \langle x_i, x_j \rangle_J) \\ \text{where } \|u\|_J =& \sqrt{u^T J u}, \quad \langle u, v \rangle_J = u^T J v \end{aligned} \tag{9}$$

where h_i and h_j are scalar node features, $\|\cdot\|_J^2$ and $\langle\cdot\rangle_J^2$ are norms and inner products computed with metric tensor J and ϕ_e is a neural network. The tensor J can be varied to enforce different symmetries, such as $\mathrm{diag}(1, -1, -1, -1)$ for $\mathrm{O}(1, 3)$ and Id_n for $\mathrm{E}(n)$. Under this formulation, the input features for ϕ_e are group invariant scalars, which leads to equivariance of the entire architecture.

However, the selection of metric tensor J relies on knowledge of the specific symmetry group, and the application of such equivariant models is restricted if no a priori knowledge about the symmetry is readily available. We show that the discovered symmetry from LieGAN can replace the requirement of theoretical knowledge through a simple procedure. First, we derive an equivalent relation between an arbitrary Lie group symmetry and its invariant metric tensor (see Appendix A.3 for proof).

Proposition 1. *Given a Lie algebra basis $\{L_i \in \mathbb{R}^{k\times k}\}_{i=1}^c$, $\eta(u, v) = u^T J v$ $(u, v \in \mathbb{R}^k, J \in \mathbb{R}^{k\times k})$ is invariant to infinitesimal transformations in the Lie group G generated by $\{L_i\}_{i=1}^c$ if and only if $L_i^T J + J L_i = 0$ for $i = 1, 2, ..., c$.*

This suggests that if we have the discovered a Lie algebra basis $\{L_i\}_{i=1}^c$, we can obtain the invariant metric tensor J for the corresponding Lie group easily by solving a linear equation. However, directly solving this system gives a zero solution for J. Also, as the basis discovered by LieGAN inevitably has some numerical error, there may not be a nonzero solution. Taking these into consideration, we add a regularization term and optimize the following objective to get an approximation of ideal metric tensor

$$\arg\min_J \sum_{i=1}^{c} \|L_i^T J + J L_i\|^2 - a \cdot \|J\|^2 \tag{10}$$

with $a > 0$. The choice of regularization coefficient a and the type of matrix norm can be flexible. A small push from zero is sufficient to get a reasonable metric J. With this approach, we can construct equivariant GNN for any discovered Lie group, which we refer to as LieGNN.

6. Experiments

We experiment on several tasks to demonstrate the capability of LieGAN. Specifically, we aim to validate (1) whether LieGAN can discover different types of symmetries mentioned in Table 1; (2) whether the discovered symmetry, combined with existing models, can boost prediction performance.

6.1. Baselines

Direct comparison with other symmetry discovery methods is not always possible, since these works deal with different settings for discovery (see Table 1). MSR (Zhou et al., 2020) uses a largely different discovery scheme from ours and can only learn finite symmetry groups, so it is not included in the experiments. SymmetryGAN (Desai et al., 2021) only learns an individual group element, which differs from our definition of symmetry discovery. We only include it in the first experiment to explain the difference. We mainly compare our method with Augerino (Benton et al., 2020), which also learns with Lie algebra representation. Augerino was originally developed for discovering a subset of a given group rather than an unknown symmetry group. We adapted Augerino from parameterizing the distribution over the given group to the distribution over the entire general linear group search space. Specifically, in the original Augerino forward function

$$f_{\mathrm{aug-eq}}(x) = \mathbb{E}_{g\sim\mu} g^{-1} f(gx), \tag{11}$$

we parameterize the distribution μ as in Equation (4). This provides ground for comparison between our method and theirs. To differentiate between this modified version with the original Augerino, we denote this approach as Augerino+ in the following discussion.

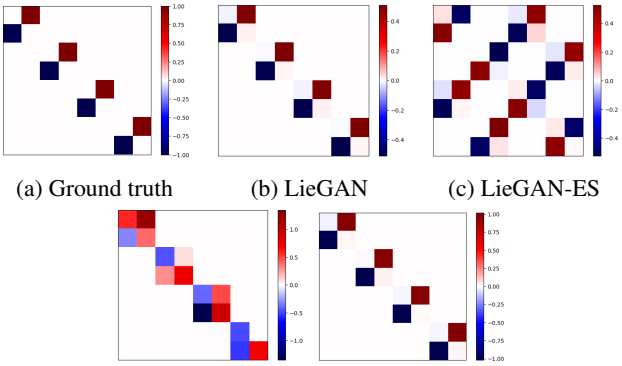
Also, we incorporate the symmetry learned by the discovery algorithms into compatible models such as EMLP (Finzi et al., 2021) and LorentzNet (Gong et al., 2022). It should be noted that these prediction models are not directly comparable with our method since they use known symmetry whereas we focus on symmetry discovery. We combine them with LieGAN to verify whether our learned symmetry representation leads to comparable prediction accuracy with the exact symmetry in theory.

6.2. N-Body Trajectory

We test our model as well as the baselines, Augerino and SymmetryGAN, on the simulated n-body trajectory dataset from Hamiltonian NN (Greydanus et al., 2019). It consists of the interdependent movements of multiple masses. We use a setting where two bodies with identical masses rotate around one another in nearly circular orbits. The task is to predict future movements based on the past series, which is rotational equivariant. The input and output feature for each timestep has $4n$ dimensions, consisting of the positions and momentums of all bodies: $[q_{1x}, q_{1y}, p_{1x}, p_{1y}, ..., q_{nx}, q_{ny}, p_{nx}, p_{ny}]$. The dataset and training details, as well as an alternative experiment setting

with three bodies, are provided in Appendix C.1.

We search for symmetries acting on the position and momentum of each mass separately, which induces a parameterization of 2×2 block diagonal matrix for the generator.



(a) Ground truth (b) LieGAN (c) LieGAN-ES

(d) Augerino+ (e) SymmetryGAN

Figure 3. Comparison between different methods on 2-body trajectory dataset. LieGAN discovers the correct rotation symmetry with both the original parameterization and the alternative one with expanded search space (LieGAN-ES), whereas Augerino+ fails. SymmetryGAN only discovers one group element.

As is shown in Figure 3, LieGAN can discover a symmetry that is nearly identical to ground truth, with a cosine correlation of 0.9998. We should note that the scale of the generator should not be taken into consideration when we compare different representations, because they are basis in the Lie algebra and are scale irrelevant. In contrast, Augerino+ only achieves a cosine similarity of 0.4880 with ground truth, which suggests that Augerino cannot be readily applied to discovering unknown groups.

On the other hand, SymmetryGAN (Desai et al., 2021) produces a very similar visualization to ground truth symmetry. However, this result has a completely different interpretation. Instead of a Lie algebra generator that generates the entire group, SymmetryGAN is learning only one element of the group. In this case, it learns a rotation by $\frac{\pi}{2}$, which coincides with the Lie algebra generator.

In addition, we expand the symmetry search space of LieGAN to enable interactions between the position or momentum of different bodies. The result is shown in Figure 3c. Given that the origin is located at the center of mass and that the two bodies have the same mass, this can be viewed as another possible representation of the same rotation symmetry. Details of derivation for this result are included in Appendix A.2.

Besides the interpretation of the learned symmetry, we can also inject it into equivariant MLP or use it augment the training data. For prediction, The train and test datasets are constructed to have different distributions so that knowledge of symmetry would be useful for generalization. The results are shown in Table 2. All experiments use the same configuration for MLP except for the introduced equivariance or data augmentation procedure. For Equivariant MLP, the two parameterizations of LieGAN outperform other symmetry discovery methods, approaching the performance of ground truth symmetry. MLP with no equivariance constraint can achieve lower training loss, but has trouble generalizing to a test set with the shifted distribution. For data augmentation, LieGAN can also achieve comparable accuracy to ground truth symmetry. SymmetryGAN only transforms the data by a fixed transformation, and its performance lies between continuous augmentation and no augmentation.

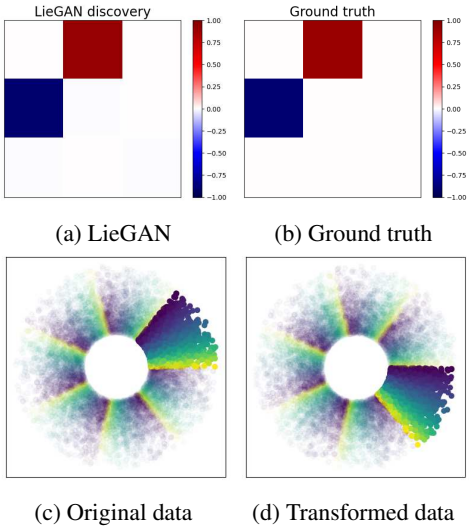
Table 2. Test MSE loss of 2-body trajectory prediction. LieGAN and LieGAN-ES correspond to different parameterizations of our model as is shown in Figure 3. Symmetries from different discovery models and ground truth are inserted into EMLP or used to perform data augmentation. HNN is also included for camparison between equivariant models and model with other types of inductive bias.

Model	EMLP	Data Aug.
LieGAN	6.43e-5	3.79e-5
LieGAN-ES	2.41e-4	6.17e-5
Augerino+	9.41e-4	1.47e0
SymmetryGAN	-	6.79e-4
Ground truth	**9.45e-6**	**1.39e-5**
HNN	3.63e-4	
MLP	8.49e-2	

6.3. Synthetic Datasets

Next, we apply LieGAN to a synthetic regression problem given by $f(x, y, z) = z/(1 + \arctan \frac{y}{x} \mod \frac{2\pi}{k})$. This function is invariant to rotations of a multiple of $2\pi/k$ in xy plane, which form a discrete cyclic subgroup of $\mathrm{SO}(2)$ with a size of k. The goal is to demonstrate that our model can capture the symmetries of not only continuous Lie groups but also their discrete subgroups.

In this task, we fix the coefficient distribution to a uniform distribution on an integer grid of $[-10, 10]$ to capture discrete symmetry. Figure 4 shows an example of LieGAN discovery in a dataset with C_7 rotation symmetry. The discovered symmetry is almost identical to ground truth, with an MAE of 0.003. Unlike the previous case of continuous rotation, the scale of the basis matters because LieGAN is modeling a set of discrete rotation symmetries with fixed angles. When acting on data, LieGAN leaves the overall data distribution unchanged while non-trivially transforms individual data points in the highlighted sector. LieGAN discovers not only the rotation group but also the correct



(a) LieGAN (b) Ground truth

(c) Original data (d) Transformed data

Figure 4. Result on the synthetic discrete rotation invariant task. (a-b): LieGAN discovers the correct rotation group and the correct scale of transformations. (c-d): Data distribution on $z = 1$ plane. The color indicates the output function value. LieGAN leaves the overall data distribution unchanged while non-trivially rotating individual data points in the highlighted sector.

scale of transformations, which demonstrates its ability to learn a subgroup of an unknown group, which is yet another generalization from discovering the continuous symmetry of an entire Lie group.

Additional results on synthetic tasks can be found in Appendix C.2 and C.3. For this rotation invariant task, we change the parameter k to show that LieGAN can capture different discrete rotation groups. We also compare LieGAN with the baseline, SymmetryGAN, to demonstrate its advantage. Besides, other synthetic functions are designed to show that LieGAN can deal with various symmetry groups and can even work well on complex values.

6.4. Top tagging

We are also interested in finding symmetry groups with more complicated structures. For example, Lorentz group is an important set of transformations in many physics problems. It is a 6-dimensional Lie group with 4 connected components. While our method cannot be readily generalized to the problem of finding discrete generators, we can test whether it is capable of extracting the identity component of the Lorentz group, $\mathrm{SO}(1,3)^+$. We use Top Quark Tagging Reference Dataset (Kasieczka et al., 2019) for discovering Lorentz symmetry, where the task is to classify between top quark jets and lighter quarks. There are 2M observations in total, each consisting the four-momentum of up to 200 particle jets. The classification task is Lorentz invariant, because a rotated or boosted input momentum should belong to the same category.

In this task, we set the generator to have up to 7 channels, which is slightly more than enough to capture the structure of 6-dimensional $\mathrm{SO}(1,3)^+$. We use cosine similarity as between-channel regularization function l_{chreg}.

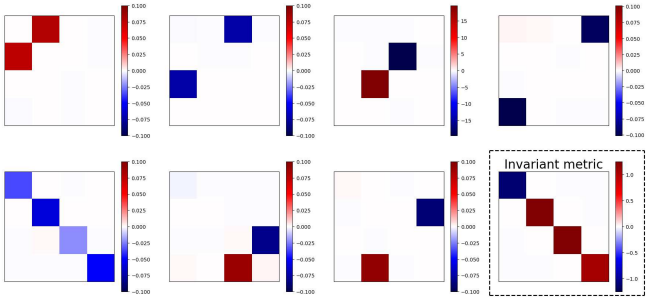
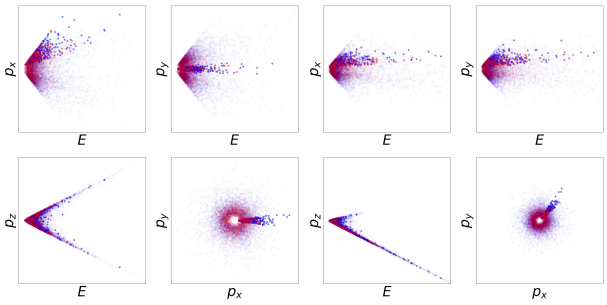


Figure 5. LieGAN discovers an approximate $\mathrm{SO}(1,3)^+$ symmetry in top tagging dataset, where channels 0, 1, 3 indicate boost along x-, y- and z-axis and channels 2, 5, 6 correspond to $\mathrm{SO}(3)$ rotation. Bottom-right: Computed invariant metric of the discovered symmetry by solving Equation (10).

The discovery results are shown in Figure 5. The four dimensions in the matrix correspond to the 4-momentum $(E/c, p_x, p_y, p_z)$. LieGAN is successful in recovering the $\mathrm{SO}(1,3)^+$ group. Its channels 2, 5, 6 correspond to $\mathrm{SO}(3)$ rotation, and channels 0, 1, 3 indicate boost along x-, y- and z-axis. In addition, the generator learns an additional Lie algebra element that scales different input dimensions with approximately the same amounts.



(a) Original (b) Transformed

Figure 6. The data distribution before and after the LieGAN transformations. The overall distribution remains unchanged, while the highlighted data points are non-trivially transformed.

Besides, figure 6 visualizes the distribution of the leading jet component in each event before and after LieGAN transformations. For better demonstration, four 2D marginal distributions of $(E, p_x), (E, p_y), (E, p_z), (p_x, p_y)$ are plotted. The overall distribution remains unchanged, while the data points in the highlighted portions are rotated and boosted

Table 3. Test accuracy and AUROC on top tagging. Our proposed model, LieGNN, reaches the performance with LorentzNet which explicitly encodes Lorentz symmetry. The result of non-equivariant GNN (LorentzNet (w/o)) and EGNN is from Gong et al. (2022).

Model	Accuracy	AUROC
LorentzNet	**0.940**	**0.9857**
LieGNN	0.938	0.9848
LorentzNet (w/o)	0.934	0.9832
EGNN	0.922	0.9760

to new locations. These results suggest that LieGAN is capable of discovering high-dimensional Lie groups and also decoupling the group structure to a simple and interpretable representation of Lie algebra basis.

It is also possible to inject this knowledge of Lie group symmetry into existing prediction models. Following the guideline in Section 5.2, we compute the invariant metric of the discovered symmetry (Figure 5 bottom-right), which is almost identical to the true Minkowski metric, with a cosine correlation of -0.9975. The computed metric is used to construct the LieGNN equivariant to the discovered group. Table 3 shows the prediction results. Without requiring any prior knowledge, LieGNN with the metric derived from LieGAN discovery reaches the performance of LorentzNet (Gong et al., 2022) with the true Minkowski metric.

7. Conclusion

In this paper, we present a method of discovering symmetry from training dataset alone with generative adversarial network. Our proposed framework addresses the discovery of various symmetries, including continuous Lie group symmetries and discrete subgroup symmetries, which is a significant step forward compared to existing symmetry discovery methods with relatively narrow search space for symmetry. We also develop pipelines for utilizing the learned symmetry in downstream prediction tasks through equivariant model and data augmentation, which proves to improve prediction performance on a variety of datasets.

This work currently deals with global symmetry of subgroups of general linear groups. However, it is also possible to apply this framework to more general scenario of symmetry discovery, such as non-connected Lie group symmetry, nonlinear symmetry and gauge symmetry, by replacing the simple linear transformation generator in LieGAN with more sophisticated structure. For instance, nonlinear symmetry could possibly be found by adding layers in generator to project the input to a space with linear symmetry.

Moreover, LieGAN shows tremendous potential in its application to supervised prediction tasks, which suggests that automatic symmetry discovery methods may eventually replace the need of human prior knowledge about symmetry. However, this ultimate vision can be fully realized only if equivariant neural network models can be implemented for more general choices of symmetry groups rather than a few specific symmetries. We have demonstrated in this work how to incorporate the discovered symmetry into some equivariant models including EMLP and EGNN, which we hope could inspire further exploration in this topic.

Acknowledgement

This work was supported in part by the U.S. Department Of Energy, Office of Science, U. S. Army Research Office under Grant W911NF-20-1-0334, Google Faculty Award, Amazon Research Award, and NSF Grants #2134274, #2107256 and #2134178.

References

Alet, F., Doblar, D., Zhou, A., Tenenbaum, J., Kawaguchi, K., and Finn, C. Noether networks: meta-learning useful conserved quantities. *Advances in Neural Information Processing Systems*, 34:16384–16397, 2021.

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Bekkers, E. J. B-spline cnns on lie groups. *arXiv preprint arXiv:1909.12057*, 2019.

Benton, G., Finzi, M., Izmailov, P., and Wilson, A. G. Learning invariances in neural networks from training data. *Advances in neural information processing systems*, 33: 17605–17616, 2020.

Blum, L. C. and Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25):8732–8733, 2009.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pp. 1321–1330. PMLR, 2019a.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Cohen, T. S., Geiger, M., and Weiler, M. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019b.

Dao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., and Ré, C. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pp. 1528–1537. PMLR, 2019.

Dehmamy, N., Walters, R., Liu, Y., Wang, D., and Yu, R. Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 34:2503–2515, 2021.

Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Desai, K., Nachman, B., and Thaler, J. Symmetrygan: Symmetry discovery with deep learning. *arXiv preprint arXiv:2112.05722*, 2021.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3244–3253. PMLR, 2019.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pp. 3165–3176. PMLR, 2020.

Finzi, M., Welling, M., and Wilson, A. G. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pp. 3318–3328. PMLR, 2021.

Gong, S., Meng, Q., Zhang, J., Qu, H., Li, C., Qian, S., Du, W., Ma, Z.-M., and Liu, T.-Y. An efficient lorentz equivariant graph neural network for jet tagging. *Journal of High Energy Physics*, 2022(7), jul 2022. doi: 10.1007/jhep07(2022)030. URL https://doi.org/10.1007%2Fjhep07%282022%29030.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, pp. 2672–2680, 2014.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2016.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Kasieczka, G., Plehn, T., Thompson, J., and Russel, M. Top quark tagging reference dataset, March 2019. URL https://doi.org/10.5281/zenodo.2603256.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Knapp, A. W. and Knapp, A. W. *Lie groups beyond an introduction*, volume 140. Springer, 1996.

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pp. 2747–2755. PMLR, 2018.

Krippendorf, S. and Syvaeri, M. Detecting symmetries with neural networks. *Machine Learning: Science and Technology*, 2(1):015010, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Romero, D. W. and Lohit, S. Learning partial equivariances from data. *arXiv preprint arXiv:2110.10211*, 2021.

Rupp, M., Tkatchenko, A., Müller, K.-R., and Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Wang, R., Walters, R., and Yu, R. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wta_8Hx2KD.

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. *arXiv preprint arXiv:2201.11969*, 2022.

Weiler, M. and Cesa, G. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

Worrall, D. and Welling, M. Deep scale-spaces: Equivariance over scale. *Advances in Neural Information Processing Systems*, 32, 2019.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Zhou, A., Knowles, T., and Finn, C. Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*, 2020.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A. Proofs

A.1. Optimizng the GAN loss function

We show in this section how optimizing the GAN loss function can lead to proper symmetry discovery. We assume in the first place the existence of a symmetry group and derive the properties of loss when the generator learns this group or its subgroup. We use the definition of perfect symmetry, that is, $gf(x) = f(gx)$ or $P_d(gf(x)|gx) = 1$ for symmetry group elements g. We use p_d and p_{gen} to denote the original distribution of data and the generated distribution.

Assumption 1. There exists a maximal subgroup of $GL(n)$, denoted as G^*, which $y = f(x)$ is equivariant to. That is, $\forall g \in G^*, gy = f(gx); \forall g \in GL(n) \backslash G^*, p_d(gy \neq f(gx)) > 0$.

Theorem 1. *The generator can achieve zero JS divergence by learning a maximal subgroup $G^* \subset \mathrm{GL}(n)$ with respect to which $y = f(x)$ is equivariant if $p_d(x)$ is distributed proportionally to the volume of inverse group element transformation along each orbit of G^*-action on $\mathcal{X}$, that is, $p_d(gx_0) \propto |\rho_{\mathcal{X}}(g^{-1})||\rho_{\mathcal{Y}}(g^{-1})|$.*

Proof. Revisiting Eq (3), the generated distribution is given by

$$p_{gen}(x, y) = \int_{G^*} \mu(g) p_d(g^{-1}x) p_d(g^{-1}y|g^{-1}x) / |\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)| dg \tag{12}$$

If $p_d(x)$ is proportionally distributed along each orbit of G^*-action on $\mathcal{X}$, then

$$p_{gen}(x, y) = \int_{G^*} \mu(g) p_d(x) p_d(g^{-1}y|g^{-1}x) dg \tag{13}$$

For any group element $g \in G^*$, $p_d(y|x) = p_d(g^{-1}y|g^{-1}x)$. Therefore,

$$p_{gen}(x, y) = (\int_{G^*} \mu(g) dg) p_d(x) p_d(y|x) \tag{14}$$

$$= p_d(x, y) \tag{15}$$

As this equality holds for all (x, y), we have zero divergence between these two distributions, p_d and p_{gen}. □

While this distribution condition is often not satisfied in practice, we further show that under certain assumptions on data and an ideal discriminator, a nontrivial Lie subgroup of the true symmetry group corresponds to a local minimum of generator loss function.

Assumption 2. For each datapoint from the original distribution, transformations outside that maximal subgroup G_T on it would not produce a valid datapoint. Formally, denoting $\bar{G^*} = GL(n) \backslash G^*$, $\int_{\bar{G^*}} \mu(g) p_d(gy|gx) dg = 0$. (While there might be slim chances that $gf(x) = f(gx)$ for some $g \in \bar{G^*}$, the integration can still yield zero as long as we parameterize $\mu(g)$ with good properties.)

Assumption 3. For each orbit $[x]$ of G^* with $P_d([x]) > 0$, $\exists x_0 \in [x], c > 0, m > 0$ s.t. $\forall g \in \delta_0(m)$, $p_d(gx_0) \geq c$, $p_d(g^2 x_0) \geq c$, and $V(g) = |\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)| \in (v_m, V_m)$, where $\delta_0(m)$ is a neighborhood of id with $P_\mu(\delta_0(m)) = m$ and $V_m > v_m > 0$ are constants depending on m.

This is actually a much more relaxed version of distribution constraint along the group action orbits in Theorem 1, which may be unrealistic. We assume instead that there exists a continuous neighborhood in each orbit where the density of x is above some threshold.

Theorem 2. *Under assumptions 1, 2 and 3, the GAN loss function under the ideal discriminator $L(\Phi, D^*)$ is lower with a generator that learns a subspace of the true Lie algebra $\mathfrak{g}^*$ than a generator with an orthogonal Lie algebra to $\mathfrak{g}^*$. That is, if $\mathfrak{g}_1 \cap \mathfrak{g}^* \neq \{\mathbf{0}\}$, $\mathfrak{g}_2 \cap \mathfrak{g}^* = \{\mathbf{0}\}$, then $L(\mathfrak{g}_1, D^*) < L(\mathfrak{g}_2, D^*) = 0$.*

Proof. As an established result in GAN, the optimal discriminator for the loss function (2) is

$$D^*(x, y) = \frac{p_d(x, y)}{p_d(x, y) + p_{gen}(x, y)} \tag{16}$$

Substituting (16) into (2), we get

$$L(\Phi, D^*) = \int p_d(x,y) \log \frac{p_d(x,y)}{p_d(x,y)+p_{gen}(x,y)} + p_{gen}(x,y) \log \frac{p_{gen}(x,y)}{p_d(x,y)+p_{gen}(x,y)} dxdy \tag{17}$$

$$= \int_{p_d(x,y)\neq 0} p_d(x,y) \log \frac{p_d(x,y)}{p_d(x,y)+p_{gen}(x,y)} + p_{gen}(x,y) \log \frac{p_{gen}(x,y)}{p_d(x,y)+p_{gen}(x,y)} dxdy \tag{18}$$

where, denoting $\tilde{\mu}(g) = \mu(g)/|\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)|$,

$$p_{gen}(x,y) = \int_g \tilde{\mu}(g) p_d(g^{-1}x) p_d(g^{-1}y|g^{-1}x) dg \tag{19}$$

Because the Haar measure dg is invariant to inversion, we have

$$p_{gen}(x,y) = \int_g \tilde{\mu}(g^{-1}) p_d(gx) p_d(gy|gx) dg \tag{20}$$

In practice, we use Gaussian distribution for $\mu(g)$, which assigns the same probability for a group element and its inverse. (This is also true for many other common choices of distribution, such as uniform distribution centered at origin.) Therefore, denoting $V(g) = |\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)|$,

$$p_{gen}(x,y) = \int_g \mu(g) p_d(gx) p_d(gy|gx) |\rho_{\mathcal{X}}(g)||\rho_{\mathcal{Y}}(g)| dg \tag{21}$$

$$= \int_g \mu(g) p_d(gx) p_d(gy|gx) V(g) dg \tag{22}$$

It is easy to show that the Lie group generated by the intersection of two Lie algebras coincides with the intersection of Lie groups generated by these two Lie algebras, respectively. Therefore, as $\mathfrak{g}_2 \cap \mathfrak{g}_T = \{\mathbf{0}\}$, $G_2 \cap G_T = \{\mathrm{id}\}$. According to Assumption 2,

$$p_{gen}(x,y;G_2) = \int_g \mu(g) p_d(gx) p_d(gy|gx) V(g) dg \tag{23}$$

$$= \int_{g\in\delta_0(1-\eta)} \mu(g) p_d(gx) p_d(gy|gx) V(g) dg + \int_{g\notin\delta_0(1-\eta)} \mu(g) p_d(gx) p_d(gy|gx) V(g) dg \tag{24}$$

$$\leq \int_{g\in\delta_0(1-\eta)} M\mu(g) p_d(gy|gx) dg + \int_{g\notin\delta_0(1-\eta)} \mu(g) p_d(gx) V(g) dg \tag{25}$$

$$= 0 + \int_{g\notin\delta_0(1-\eta)} \mu(g) p_d(gx) V(g) dg \tag{26}$$

where, following the notations in Assumption 3, $\delta_0(1-\eta)$ is the neighborhood of id, $P_\mu(\delta_0(1-\eta)) = 1-\eta$, and $V(g) \leq V_{1-\eta}$. Therefore, there exists an upper bound $M = \max_{g\in\delta_0(1-\eta)} p_d(gx)V(g)$.

For the integral on $g \notin \delta_0(1-\eta)$, as the Gaussian density $\mu(g)$ decays exponentially with $V(g)$ and $p_d(gx)$ has an upper bound, $\forall \epsilon > 0, \exists \eta$ s.t. $\int_{g\notin\delta_0(1-\eta)} \mu(g) p_d(gx) V(g) dg < \epsilon$.

Therefore, $p_{gen}(x,y;G_2) = 0$ and $L(\mathfrak{g}_2, D^*) = 0$.

On the other hand, $\mathfrak{g}_1 \cap \mathfrak{g}^* \neq \{\mathbf{0}\} \Rightarrow G_1 \cap G^* \neq \{\mathrm{id}\}$. We consider the integral (18) along each possible orbit of G^*. According to Assumption 3, there exists an x-neighborhood $X = \delta_0(m)x_0$ s.t. $\forall x \in X, p_d(x, f(x)) > c$. For the generated

distribution on this neighborhood, we have

$$p_{gen}(x, f(x)) = \int_g \mu(g) p_d(gx) V(g) dg \tag{27}$$

$$\geq \int_{g \in \delta_0(m)} \mu(g) p_d(gx) V(g) dg \tag{28}$$

$$\geq \int_{g \in \delta_0(m)} \mu(g) c v_m dg \tag{29}$$

$$= m c v_m > 0 \tag{30}$$

As the supports of p_d and p_{gen} overlap on this neighborhood, we have $L(\mathfrak{g}_1, D^*) < 0 = L(\mathfrak{g}_2, D^*)$. □

A.2. Experiment Result on 2-Body Trajectory Dataset

In Figure 3c, we observe an unfamiliar symmetry representation. In fact, this is another possible representation for rotation symmetry. The learned Lie algebra basis L can be expressed in the following form after discarding the noise:

$$R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} R & & -R & \\ & R & & -R \\ -R & & R & \\ & -R & & R \end{bmatrix}$$

Computing the matrix exponential gives

$$\exp(\theta L) = \begin{bmatrix} L(\theta) & & -L(\theta) & \\ & L(\theta) & & -L(\theta) \\ -L(\theta) & & L(\theta) & \\ & -L(\theta) & & L(\theta) \end{bmatrix} + I$$

$$L(\theta) = \sum_{k=0}^{+\infty} \frac{2^{2k}(-1)^k \theta^{2k+1}}{(2k+1)!} R + \sum_{k=1}^{+\infty} \frac{2^{2k-1}(-1)^k \theta^{2k}}{(2k)!} I$$

As the origin for this dataset is at the center of mass and $m_1 = m_2$, we have $\mathbf{q}_1 = -\mathbf{q}_2$ and $\mathbf{p}_1 = -\mathbf{p}_2$. Therefore,

$$\exp(\theta L) \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \\ \mathbf{q}_2 \\ \mathbf{p}_2 \end{bmatrix} = \mathrm{diag}(I + 2L(\theta)) \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \\ \mathbf{q}_2 \\ \mathbf{p}_2 \end{bmatrix}$$

$$I + 2L(\theta) = \sum_{k=0}^{+\infty} \frac{2^{2k+1}(-1)^k \theta^{2k+1}}{(2k+1)!} R + \sum_{k=0}^{+\infty} \frac{2^{2k}(-1)^k \theta^{2k}}{(2k)!} I$$

$$= \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

which indicates that this is another representation for rotation specific to this dataset.

A.3. Computing Group Invariant Metric Tensor

Proposition 1. *Given a Lie algebra basis $\{L_i \in \mathbb{R}^{k \times k}\}_{i=1}^c$, $\eta(u, v) = u^T J v$ ($u, v \in \mathbb{R}^k, J \in \mathbb{R}^{k \times k}$) is invariant to infinitesimal transformations in the Lie group G generated by $\{L_i\}_{i=1}^c$ if and only if $L_i^T J + J L_i = 0$ for $i = 1, 2, ..., c$.*

Proof. An infinitesimal transformation in group G generated by the given Lie algebra basis can be written as the matrix

representation $g = I + \sum_i \epsilon_i L_i$.

$$\eta(u, v) = \eta(gu, gv) \tag{31}$$
$$\iff u^T J v = u^T g^T J g v \tag{32}$$
$$\iff u^T (I + \sum_i \epsilon_i L_i^T) J (I + \sum_i \epsilon_i L_i) v = u^T J v \tag{33}$$
$$\iff u^T J v + u^T (\sum_i \epsilon_i (L_i^T J + J L_i)) v + O(\epsilon^2) = u^T J v \tag{34}$$
$$\iff u^T (\sum_i \epsilon_i (L_i^T J + J L_i)) v = 0, \forall u, v \in \mathbb{R}^n \tag{35}$$

As this holds for any infinitesimal transformation g, we can set $\epsilon_{-i} = 0$ to get

$$\epsilon_i (L_i^T J + J L_i) = 0, i = 1, 2, ..., c \tag{36}$$

Therefore, $L_i^T J + J L_i = 0, i = 1, 2, ..., c$.

On the other hand, if $L_i^T J + J L_i = 0, i = 1, 2, ..., c$, then

$$\sum_i \epsilon_i (L_i^T J + J L_i) = 0, \forall \epsilon \in \mathbb{R}^c \tag{37}$$

□

B. Experiment Details

This section provides detailed explanation on the experiment settings, including the dataset generating procedure, the hyperparameters used in training, etc.

B.1. N-Body Trajectory

We use the code from Hamiltonian Neural Networks [1] (Greydanus et al., 2019) to generate the dataset for this task. We construct the train and test sets with different distributions to test the generalization ability of the models. Specifically, we sort the samples in terms of the polar angle of the position of the first particle at the starting timestep of trajectory, and divide the sorted dataset into train and test sets.

The task for this dataset is to predict K future timesteps of 2-body movement based on P past timesteps of observation, where the feature for each timestep has 8 dimensions, consisting of the positions and momentums of two bodies: $[q_{1x}, q_{1y}, p_{1x}, p_{1y}, q_{2x}, q_{2y}, p_{2x}, p_{2y}]$. In our experiment, we set $P = K = 5$. When discovering symmetry, the LieGAN generator takes both the past observations and future predictions as input, yielding an input dimension of 80. The generator transforms each timestep at the same time, which means that it is learning a group representation of $\mathbb{R}^{8\times 8}$ that acts simultaneously on each of the past input and future output timesteps. On the other hand, we use a 3-layer MLP as with discriminator, with input dimension 80, hidden dimension 512, and leaky ReLU activation with negative slope 0.2. We use only the regularization against identical transformations, i.e. $l_{\text{reg}}(x, y)$ in (5), with the regularization coefficient $\lambda = 1$. but not the between-channel regularization in (6), because the generator only has a single channel and there is no need for it. The learning rates for the discriminator and the generator are set to 0.0002 and 0.001, respectively. LieGAN is trained adversarially for 100 epochs.

In the prediction task with equivariant model, we use EMLP with 3 hidden layers and a hidden representation of $5V$, where V stands for an 8-dimensional vector just as the feature for each timestep. We train all EMLPs constructed with different equivariances with lr=0.0001 for 5000 epochs.

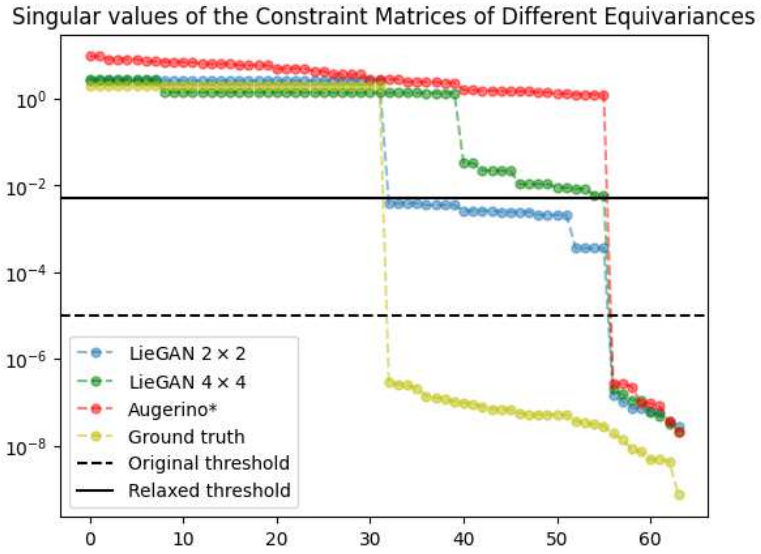


Figure 7. Singular values of the EMLP constraint matrices derived from different equivariances under the representation of group actions on weight matrices mapping from $V_1 \to V_2$, where V_1 and V_2 are both 8-dimensional vector spaces. The y axis is log-scaled for better visualization. It can be observed that the singular values of the constraint matrix corresponding to the symmetry discovered by LieGAN have a sharp decrease at the same position as the matrix for ground truth symmetry. This suggests that we can slightly relax the singular value threshold to obtain a higher dimensional equivariant subspace.

As is mentioned in Section 5.2, we slightly modified the EMLP implementation to adapt to the noised discovery result from LieGAN. EMLP projects the network weight to an equivariant subspace, which is the null space of the constraint

[1] https://github.com/greydanus/hamiltonian-nn

matrix derived from the provided equivariance and input and output representations. The null space is computed with SVD. This method usually works for common handpicked symmetries, such as Euclidean group and Lorentz group, which typically have sparse and clean matrix representations. However, the symmetry discovered by LieGAN inevitably has some numerical error. While such error could be largely negligible when we visualize the discovered symmetry or use it for data augmentation, it will cause problem in the SVD procedure in the EMLP implementation. Even a small noise that changes a matrix representation entry from zero to small nonzero values could result in a constraint matrix with higher rank, which then leads to a lower dimensional equivariant subspace and a lower rank weight matrix. However, we can raise the singular value threshold to larger values to calculate an approximate null space, which has higher dimensions. Figure 7 shows how we modify the singular value threshold. The original EMLP implementation sets a threshold of 1e-5. With this threshold, the symmetries discovered by LieGAN lead to a weight matrix that maps each input vector to each hidden vector with a rank of 8, significantly lower than 32, which is the case for ground truth rotation symmetry. However, it can also be observed that the singular values of the constraint matrix corresponding to LieGAN symmetry have a sudden fall at the same position as the matrix for ground truth symmetry. Therefore, we can raise the singular value threshold to 5e-3, which is still reasonably small, and obtain a 32-dimensional approximately equivariant subspace for the discovered symmetry. This procedure proves to significantly improve prediction performance for EMLP constructed with the discovered symmetry.

B.2. Synthetic Regression

This is a regression problem given by $f(x, y, z) = z/(1 + \arctan \frac{y}{x} \mod \frac{2\pi}{k})$. This function is invariant to rotations of a multiple of $2\pi/k$ in xy plane, which form a discrete cyclic subgroup of $\mathrm{SO}(2)$ with a size of k. In our experiment, we construct the dataset with $k = 7$. We randomly sample 20000 inputs (x, y, z) from a standard multivariate Gaussian distribution and calculates the output analytically. For symmetry discovery, we use a generator with a single channel of $\mathbb{R}^{3\times3}$ matrix representation and a 3-layer MLP discriminator with input dimension 4 (which is (x, y, z, f)), hidden dimension 512, and leaky ReLU activation with negative slope 0.2. The coefficient distribution in the generator is set to a uniform distribution on integer grid between $[-10, 10]$. We use regularization term l_{reg} with coefficient $\lambda = 0.01$. The learning rates for the discriminator and the generator are set to 0.0002 and 0.001, respectively. LieGAN is trained for 100 epochs.

B.3. Top Quark Tagging

For symmetry discovery, we use a generator with 7 channels of $\mathbb{R}^{4\times4}$ matrix representations acting on the input 4-momenta $(E/c, p_x, p_y, p_z)$. The input consists of the momenta of up to 200 constituents for each sample, sorted by the transverse momentum of each constituents. We truncate the input to the momenta of the two leading constituents, which gives an input dimension of 8. As this classification task is invariant, the generator does not change the category label associated with each sample. The discriminator takes both the transformed input momenta gx and the output label $gy = y$ as its input. It first transforms y to a real-valued vector with an embedding layer, and then concatenates the embedding with gx, and passes them through a 3-layer MLP with hidden dimension 512 and leaky ReLU activation with negative slope 0.2. We use regularizations l_{reg} with coefficient $\lambda = 1$ and l_{chreg} with coefficient $\eta = 0.1$. The learning rates for the discriminator and the generator are set to 0.0002 and 0.001, respectively. LieGAN is trained for 100 epochs.

For prediction with LieGNN, we first calculate the invariant metric tensor based on the discovered symmetry according to Equation (10). We optimize the objective with $a = 0.0005$ and matrix max norm using a gradient descent optimizer with step size of 1×10^{-5}. Then, we build the LieGNN prediction model based on LorentzNet implementation [2]. The model has 6 group equivariant blocks with 72 hidden dimensions. We use a dropout rate of 0.2 and weight decay rate of 0.01. The model is trained for 35 epochs with a learning rate of 0.0003. These settings are the same for LorentzNet and LieGNN.

[2] https://github.com/sdogsq/LorentzNet-release

C. Additional Experiments

C.1. N-Body Trajectory

We extend the 2-body setting in Section 6.2 to 3-body movements. Despite the increased complexity, LieGAN is still able to discover the rotation symmetry in this case, as is shown in Figure 8.

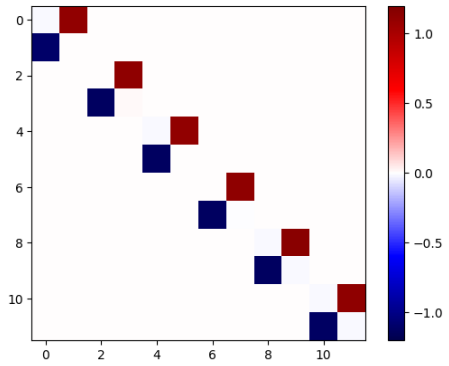


Figure 8. Symmetry discovery result on 3-body trajectory prediction dataset. LieGAN can also learn an accurate representation of rotation symmetry as in the case of 2-body dataset.

C.2. Synthetic Regression

Consider the function $f(x, y, z) = z/(1 + \arctan \frac{y}{x} \mod \frac{2\pi}{k})$ introduced in Section 6.3. We use different values of k to construct functions that are invariant to different groups. Table 4 shows the results for $k = 6, 7, 8$, corresponding to the cyclic groups $\mathrm{C}_6, \mathrm{C}_7, \mathrm{C}_8$. LieGAN successfully captures these discrete rotation groups. SymmetryGAN works fine in some cases, but its convergence heavily depends on random initialization. For example, it does not converge for $k = 8$ and converges to a non-generator element $R(4\pi/3)$ for $k = 6$.

k	6	7	8
LieGAN	0.012	0.003	0.011
SymmetryGAN	0.024*	0.034	N/A

Table 4. Mean absolute error between the discovered symmetry representations and ground truths.

C.3. More Synthetic Tasks

Partial permutation symmetry. Consider the function $f(x) = x_1 + x_2 + x_3 + x_4^2 - x_5^2,\ x \in \mathbb{R}^5$. It has partial permutation symmetry, i.e. the output stays the same if we permute the first 3 dimensions of x, but it will change if we also permute the last 2 dimensions. As permutation is a discrete symmetry, we set the coefficient distribution to a uniform distribution on an integer grid, similar to the setting in Section 6.3.

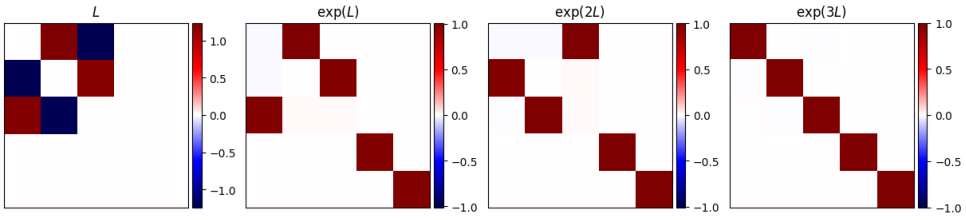


Figure 9. Discovered partial permutation symmetry.

Figure 9 shows the discovery result. The LieGAN generator exactly matches the ground truth,

$$L_{\text{truth}} = \log\left(\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}\right) \approx 1.21 \times \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with $\text{MAE} = 0.003$. The figure also shows that when we compute the exponential of L, $2L$ and $3L$, we get the permutations (123), (132) and id.

$\text{SU}(2)$ **symmetry.** In this example, we show that LieGAN can also work on complex-valued tasks. Consider the function $f(x, y) = \frac{1}{2}(x_1 y_2 - x_2 y_1)^2 + (x_1 y_2 - x_2 y_1),\ x, y \in \mathbb{C}^2$. Such holomorphic functions are referred to as superpotentials which are relevant to supersymmetric field theories (Krippendorf & Syvaeri, 2020). We want to find a complex Lie algebra representation, i.e. $\{L_i \in \mathbb{C}^{2\times 2}\}_{i=1}^{c}$, that acts on the inputs, x and y, simultaneously. The true underlying invariance here is the special unitary group, $\text{SU}(2)$.

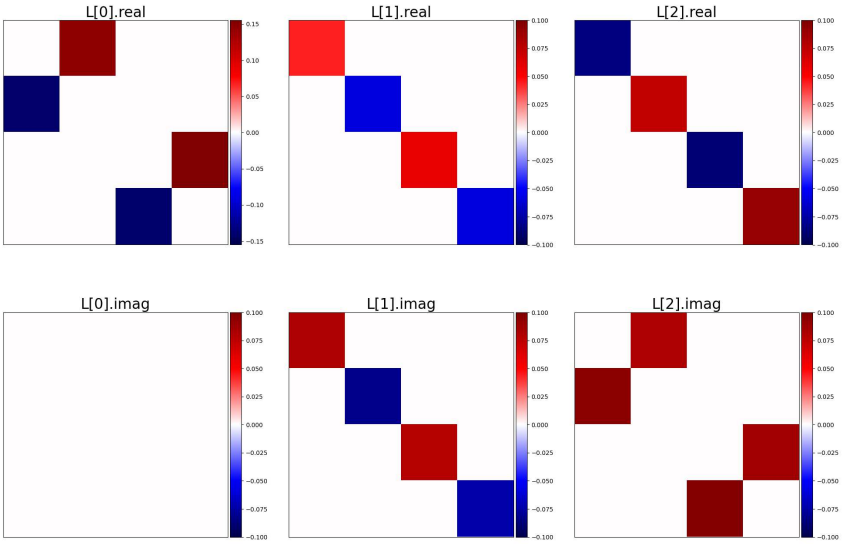


Figure 10. The complex Lie algebra discovered by LieGAN.

We set the number of generator channels to $c = 3$. Figure 10 shows the discovered complex Lie algebra. It can be approximately written in the following numerical form:

$$L_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},\ L_2 = \begin{bmatrix} 1 + c_1 i & 0 \\ 0 & -1 - c_1 i \end{bmatrix},\ L_3 = \begin{bmatrix} -1 & c_2 i \\ c_2 i & 1 \end{bmatrix}$$

A more familiar form of $\mathfrak{su}(2)$ representation is given by

$$u_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},\ u_2 = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix},\ u_3 = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$$

It can be easily checked that our discovery result is equivalent to this representation upon a change of basis:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -1 & & \\ & \frac{i}{1+c_1 i} & \\ & \frac{1}{c_2(1+c_1 i)} & \frac{1}{c_2} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix}$$

Thus, we may conclude that LieGAN can identify the $\text{SU}(2)$ invariance in this function.

C.4. Rotated MNIST

We consider the classic example of image classification on the MNIST (Deng, 2012) dataset. The original dataset is transformed by rotations by up to 45 degrees so that it has $\mathrm{SE}(2)$ symmetry which includes the rotations and translations on 2D grids. We set the number of generator channels to $c = 1$. We set the search space to all affine transformations on 2D space, which is 6-dimensional. The discovery result is

$$L = \begin{bmatrix} 0.01 & -0.66 & 0.08 \\ 0.66 & -0.01 & -0.01 \\ 0 & 0 & 0 \end{bmatrix}$$

This can be interpreted as a mixture of rotation ($L[1,0] = -L[0,1] = 0.66$) and translation ($L[0,2] = 0.08$), where the magnitude of rotations is larger than the magnitude of translations. Figure 11 visualizes the original and transformed MNIST digits.

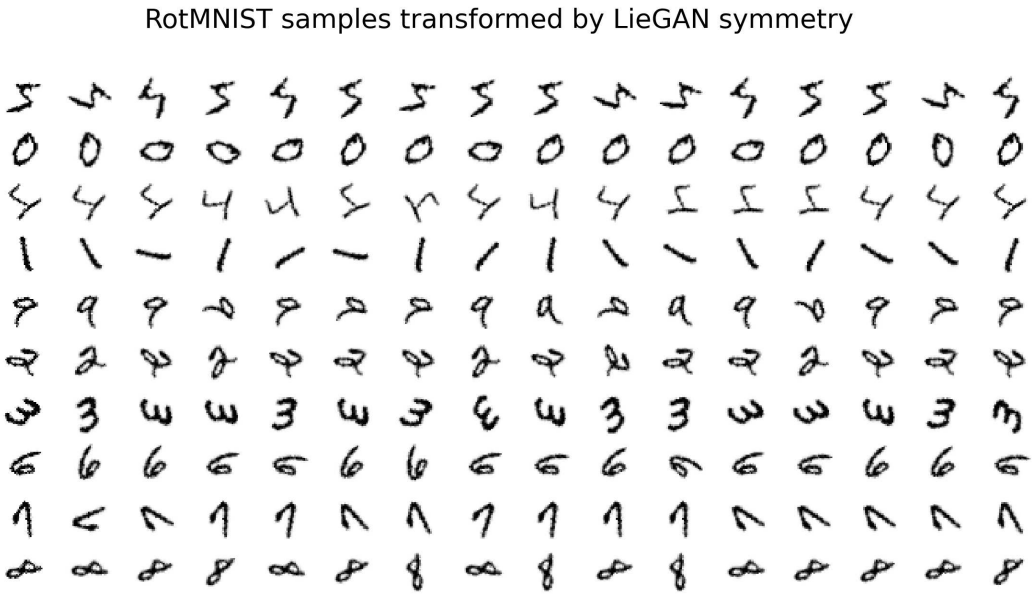


Figure 11. MNIST samples transformed by LieGAN. The first column shows the original samples from RotMNIST. For each image, we sample 15 group elements from LieGAN and plot the transformed images.

C.5. Molecular Property Prediction

We are also interested in whether LieGAN can recover $\mathrm{SE}(3)$ symmetry (the rotations and translations in 3D space), which has wide applications in computer vision, molecular dynamics, etc. Thus, we experiment on QM9 (Blum & Reymond, 2009; Rupp et al., 2012), where the task is to predict molecular properties based on the 3D coordinates and charges of the atoms. We set the number of generator channels to $c = 6$, which matches the dimension of $\mathrm{SE}(3)$. Figure 12 shows the discovered

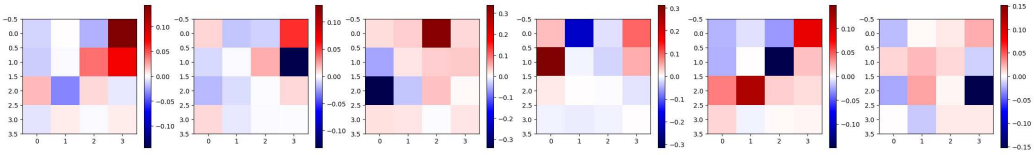


Figure 12. Discovery result for QM9 dataset.

Lie algebra representations, which produce group representations that act on the affine coordinates $(x, y, z, 1)$. LieGAN can discover an approximate $\mathrm{SE}(3)$ symmetry, where the skew-symmetric entries in the first three dimensions indicate rotations along different axes, and the non-zero entries in the last column indicate translations along different directions.

We can also use the discovered LieGAN symmetry to perform data augmentation during training. The discovered symmetry proves to increase the prediction accuracy on different QM9 tasks compared to a model with no symmetry, as is shown in table 5.

Task	No symmetry	LieGAN	$\mathrm{SE}(3)$
HOMO	52.7	43.5	**36.5**
LUMO	43.5	36.4	**29.8**

Table 5. Test MAE (in meV) on QM9 tasks. The results for no symmetry and $\mathrm{SE}(3)$ symmetry are referred from Benton et al. (2020).