Relational Macrostate Theory for Understanding and Designing Complex Systems

by

Yanbo Zhang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved August 2023 by the
Graduate Supervisory Committee:

Sara Walker, Chair
Ariel Anbar
Bryan Daniels
Jnaneshwar Das
Paul Davies

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

Scientific research encompasses a variety of objectives, including measurement, making predictions, identifying laws, and more. The advent of advanced measurement technologies and computational methods has largely automated the processes of big data collection and prediction. However, the discovery of laws, particularly universal ones, still heavily relies on human intellect. Even with human intelligence, complex systems present a unique challenge in discerning the laws that govern them. Even the preliminary step, system description, poses a substantial challenge. Numerous metrics have been developed, but universally applicable laws remain elusive. Due to the cognitive limitations of human comprehension, a direct understanding of big data derived from complex systems is impractical. Therefore, simplification becomes essential for identifying hidden regularities, enabling scientists to abstract observations or draw connections with existing knowledge. As a result, the concept of macrostates – simplified, lower-dimensional representations of high-dimensional systems – proves to be indispensable. Macrostates serve a role beyond simplification. They are integral in deciphering reusable laws for complex systems. In physics, macrostates form the foundation for constructing laws and provide building blocks for studying relationships between quantities, rather than pursuing case-by-case analysis. Therefore, the concept of macrostates facilitates the discovery of regularities across various systems. Recognizing the importance of macrostates, I propose the relational macrostate theory and a machine learning framework, MacroNet, to identify macrostates and design microstates. The relational macrostate theory defines a macrostate based on the relationships between observations, enabling the abstraction from microscopic details. In MacroNet, I propose an architecture to encode microstates into macrostates, allowing for the sampling of microstates associated with a specific macrostate. My experiments on simulated systems demonstrate the effectiveness of this theory and method in identifying macrostates such as energy. Furthermore, I apply this theory and method to a complex chemical system, analyzing oil droplets with intricate movement patterns in a Petri dish, to answer the question, “which combinations of parameters control which behavior?” The macrostate theory allows me to identify a two-dimensional macrostate, establish a mapping between the chemical compound and the macrostate, and decipher the relationship between oil droplet patterns and the macrostate.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my parents for their unwavering support throughout my Ph.D. journey.

My sincere thanks also goes to my Ph.D. advisor, Prof. Sara Walker. Her encouragement during my diverse explorations, as well as her tolerance of my initial "random walk" phase, was pivotal in bringing this dissertation to fruition.

I must extend my appreciation to my collaborators, Prof. Lee Cronin and Prof. Jnaneshwar Das. Their invaluable discussions and inspirations significantly enriched this work.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Abstract

The identification of laws in science is often connected with the discovery of invariants and symmetries. A prime example of this relationship is the concept of energy conservation. By recognizing and utilizing the concept of energy, scientists can move beyond isolated, case-by-case studies to develop general laws governing physical systems. In this context, we argue that macrostates, defined as specific subsets within a system's subspace, provide a more versatile framework for understanding complex systems in a unified manner, rather than studying different systems in isolation. This chapter offers an overview of macrostates as studied within the domain of physics. Additionally, we review the theoretical examination of macrostates from a computational standpoint. A review of practical methods for identifying macrostates or related concepts, particularly within the machine learning domain, will be presented. Finally, we propose an integrative perspective that connects various macrostate theories and methodologies, offering a cohesive approach to this multifaceted subject.

1.2 Universal Laws and Macrostates

Among the most important concepts in science is that of scientific laws, identified as regularities or rules that hold universally when a given set of conditions is met. Identifying new laws allows predictions, and the ability to design new example systems consistent with those laws. New laws can be found by identifying invariant quantities that remain unchanged under some transformation. An archetypal example is energy and Hamiltonian (Goldstein, Poole, and Safko 2002): the regularity that energy is always conserved leads to the law of conservation of energy. Identifying such quantities will then allows us to predict and build systems consistent with this law (Greydanus, Dzamba, and Yosinski 2019).

There is a fundamental connection between invariants, like energy, and symmetry. This relationship was first made clear in physics by Noether (Noether 1971). Noether showed how for systems with conservative forces, every differentiable symmetry comes with a corresponding conservation law. An example is how time translation symmetry gives rise to the conservation of energy: simple harmonic oscillators conserve energy in the absence of friction, and you will observe the same oscillations if starting a clock at the first cycle as at the thousandth because the behavior is time-invariant. However, finding laws (or symmetries) for complex systems, such as biological and technological ones, has proved more challenging that requires complex methods (Jumper et al. 2021; Pathak et al. 2018; Seif, Hafezi, and Jarzynski 2021). This is because of their high dimensionality, non-linear behavior, and emergent properties. Breaking the barrier to systematically find law-like behaviors in complex systems would allow insights into the physics underlying them and like in more simple physical systems it could provide universal tools for the prediction and design of their behaviors.

Beyond invariants, macroscopic descriptions, or macrostates, are frequently encountered and serve a wider range of applications. Despite many different definitions of macrostates, to form a term basis for the subsequent study, I adopt a definition that can cover most different theories (Gömöri, Gyenis, and Hofer-Szabó 2017):

Definition 1 (Macrostate of microstates) *A macrostate of microstates repre-*

sents a subset of microstates. Continuous macrostates can be parameterized by multi-dimensional variables, which are referred to as ***macro-variables*** *or* ***macroscopic quantities****.*

Macrostates can be either discontinuous, such as in classifications, or continuous, exemplified by variables like temperature. In the context of continuous macrostates, they can be parameterized in high-dimensional spaces. For instance, the macrostates of an ideal gas are continuous and can be parameterized by a combination of pressure, volume, and temperature (P, V, T), each dimension of which is referred to as a macro-variable or a macroscopic quantity. Each macrostate is associated with an ensemble of microstates, resulting in a many-to-one mapping from microstates to macrostates. This mapping will henceforth be referred to as the "microstate-macrostate mapping", or simply the "micro-to-macro mapping". It is important to note that macrostates and microstates cannot be defined independently; their definitions are relative, thereby inherently allowing for multiple levels of representation. Higher level representations (i.e., lower-dimensional) are macrostates of lower level representations. In the following sections, I will use macrostate instead of "macrostate of microstates" for simplicity when there is no ambiguity.

In statistical physics, three quantities—pressure, volume, and temperature—are fundamental in characterizing a gas system (Landau and Lifshitz 2013). And the concept of Boltzmann's entropy also relies on the number of microstates under certain macrostates that are represented by these three macro-variables. Another example, which can be less intuitive, can be found in Newton's Laws, which rely on quantities such as mass, speed, and acceleration. These quantities, in essence, can be averages derived from objects composed of countless molecules. Consequently, these fundamental quantities often require interpretation as macroscopic descriptions in most use cases.

Viewed from a different perspective, these macroscopic quantities facilitate the practical application of laws unearthed in laboratory settings. This is significant, as microscopic details can exhibit significant variation across different objects, locations, and timeframes. In contrast, macroscopic quantities and their governing laws have the potential to remain consistent. Comparing macrostates and invariants, it becomes evident that they share a mutual definition and overlapping relationship. Invariants such as energy or momentum are undoubtedly macroscopic descriptions as they are often a summation of micro-details. However, macrostates such as speed and mass can also formulate energy and momentum, thereby imposing constraints on these macrostates through invariant laws.

To understand the complex behaviors of intricate systems, such as biological or social systems, we need both types of quantities to formulate laws. And since invariants are a special type of macrostates, my focus rests on the theory of macrostates and on identifying macrostates from observations.

Despite the fact that only a few countable studies explicitly discussed the concept of macrostates (Hoel, Albantakis, and Tononi 2013; Shalizi and Moore 2003; Gömöri, Gyenis, and Hofer-Szabó 2017), many studies have touched on this topic in various ways. These studies often use the name of latent spaces (Hinton and Salakhutdinov 2006; Kingma and Welling 2013; Higgins et al. 2017; Zhao, Song, and Ermon 2017), representation learning (Mikolov, Sutskever, et al. 2013; Mikolov, Chen, et al. 2013; He et al. 2020), dimension reduction (Jolliffe and Cadima 2016; Van der Maaten and Hinton 2008), or emergence (Hoel, Albantakis, and Tononi 2013; Hoel 2017). In the ensuing sections of this introduction, I will initially revisit theories associated with macrostates within the physics domain. Following this, I will explore recently formulated theories that expressly focus on macrostates. My survey will then turn to an examination of methods from the machine learning field that inadvertently identify certain types of macrostates or correlate with theories advanced in recent years. Finally, I will provide an integrative perspective of macrostate theories and succinctly introduce the underlying principles of my relational macrostate theory. For a more detailed discussion, please refer to chapter 3.

1.3 Macrostate in Physics

There are many important macrostates in the physics domain, including energy, temperature, pressure, averaged speed, etc. Some of them can be directly measured, hence forming the foundation of physical theories.

One of the significant functions of macrostates lies in their ability to encapsulate the complexities of physical reality, thereby offering a simplified and condensed version of it. This is largely predicated on the inherent cognitive and computational constraints of the human mind and existing technological resources, necessitating the simplification of observations for more effective comprehension and application (Hemmo and Shenker 2012). Macrostates, thus, operate as a form of compact reality where minute details are disregarded as being indistinguishable to observers.

However, the utility of the macrostate is beyond simplification. Their reusability (Shalizi and Moore 2003) form the basis for the formulation of general laws – going beyond case-by-case study, and studying general laws. Physical macrostates like temperature, volume, and pressure, can be universally applied across any ideal gas, and multiple domains beyond gas, thereby promoting the generalizability of knowledge obtained from a singular observation to multiple, diverse observations.

Indeed, it is worth noting that all of our physical laws are formulated with a certain level of disregard for detail. A common feature of these laws is their separation from initial states and boundary conditions (Pattee, Rączaszek-Leonardi, and Pattee 2012): Newton's laws don't provide any information about entities' initial speed, or the environment information. This necessitates a degree of data compression when uncovering laws from observations, further highlighting the vital role of macrostates. Notable examples emerge from the fields of biology, geology, and climate science. Systems within these domains frequently encompass a vast amount of unobservable information and exhibit intrinsic randomness, which complicates microstate-level predictions. Subsequent studies will elucidate how macrostates can be employed to understand such systems, bypassing the need to ascertain these boundary conditions.

Intrinsically, macrostates encapsulate symmetry information. As established earlier, macrostates correspond to subsets of microstates. Any transformations maintaining the microstates within the same subset do not alter the associated macrostates. These transformations represent the symmetries inherent in the macrostates. For instance, the macrostate (P, V, T) of an isolated ideal gas remains invariant under rotations, spatial translations, and temporal translations. As such, we can infer that the mapping from the gas's microstates (i.e., the position and velocity of each particle) to the macrostate contains rotational, spatial translational, and temporal translational symmetry. Therefore, the identification of macrostates aids in uncovering symmetries within mappings. A significant application of this notion relates to the concept of "more is different" (Anderson 1972). Anderson proposed that large-scale patterns may not always keep the symmetry of their underlying rules, a phenomenon widely observed and recognized as emergence. However, when we consider the rule and pattern as a mapping, identifying macrostates allows us to discover symmetries that are preserved, or not broken. Indeed, several studies, such as Lenia (Chan 2018) and the experiments on Turing patterns in Chapter 3, have demonstrated the existence of preserved symmetry within such rule-pattern mapping systems. Although the specific outcomes of patterns cannot be precisely controlled by rules, they do have some features associated with a set of rules, implying the existence of certain symmetries. Hence, this perspective fosters a new understanding of the "more is different": while many symmetries may break as a system scales up, there often remain some unbroken symmetries.

However, identifying the macrostates is no simple task, particularly in systems exhibiting emergent phenomena such as those found in biological, social, or chaotic systems. These systems often possess high-dimensional microstates, and their macrostates cannot be reduced to a mere average of their microstates. Our everyday intuitions fail in these instances, leaving us in need of innovative theories and methodologies to identify their macrostates.

1.4 Theoretical Perspective on Macrostate Theory

As outlined in the first section, macrostates can be conceptualized as subsets of a system's microstate phase space. However, this somewhat general non-informative definition does not lead to any precise methodology for determining the elements of these subsets. Two important studies aimed at addressing this issue are the causal state theory (Shalizi and Moore 2003) and the causal emergence theory (Hoel, Albantakis, and Tononi 2013). Despite their respective limitations, they provide inspiration for my own approach – the relational macrostate theory.

1.4.1 Causal State Theory

In 2003, Shalizi and Moore introduced the causal state theory (Shalizi and Moore 2003), establishing a definition of macrostates based on the relations among microstates. It is important to clarify that the term "causal" in this context denotes another layer of system representation. This term should not be understood as a reference to the recent focus on causal science (Pearl and Mackenzie 2018) or to any conventional causal inference methodologies such as Granger causality (Granger 1969). In their theory, Shalizi and Moore propose that two microstates can be considered "causally" equivalent – that is, belonging to the same macrostate – if their subsequent distributions of microstates are the same (Figure 1A). Mathematically, two microstates histories, denoted as $\overleftarrow{s}$ and $\overleftarrow{s}'$, are considered to belong to the same macrostate (represented by $\sim$) if and only if they have identical conditional distributions for their future states, symbolized as $\overrightarrow{s}$:

$$\overleftarrow{s} \sim \overleftarrow{s}' \iff P(\overrightarrow{s}|\overleftarrow{s}) = P(\overrightarrow{s}|\overleftarrow{s}'). \tag{1.1}$$

Consequently, the conserved symmetry is related to the prediction of future states. This, however, is not generalizable and can exclude some well-defined macrostates in physics. For example, given a simple harmonic oscillator, two distinct microstates $u_1 = (p_1, x_1)$ and $u_2 = (p_2, x_2)$, where p_1 and p_2 are two observations of momentum and x_1 and x_2 are the corresponding position, can have the same energy macrostate but their future microstate distributions will be different if u_1 and u_2 are not close to each other (say if, $u_1 = -u_2$). The macrostate of energy is related to the time translation symmetry (as identified by Noether), not the symmetry associated with the predictability of future states (as in causal state theory). Indeed, Shalizi and Moore were not looking for a general theory of macrostates but instead focused on the specific property of predictability of complex systems. Indeed, their approach does not provide a similarity metric for future state distributions. This may result in measurements that are either excessively sensitive or insufficiently responsive to certain information. Thus, their approach identifies predictive states for specific systems but not general laws.

1.4.2 Causal Emergence Theory

Another approach was more recently proposed in causal emergence theory (Hoel, Albantakis, and Tononi 2013), with the goal to describe causal relations at the macro level. Here, instead of using the properties of microstates, macrostates are defined by identifying relations between macrostates. Contrary to explicitly defining the equivalence of microstates, causal emergence is developed in a more implicit manner. By defining the "effective information" (EI) of a discontinuous dynamical system evolving over time, some coarse-grained, lower-dimensional models can exhibit higher effective information than models framed at the microstate level. This increase in effective information is referred to as "causal emergence." The effective information is defined as:

$$EI(S) = \frac{1}{n} \sum_{s_0 \in U^C} D_{\mathrm{KL}}((S_F|s_0), U^E), \tag{1.2}$$

where s_0 is the initial state, sampled uniformly from all possible initial states U^C.

Meanwhile, U^E represents the next-state (or effects) distribution of the system when the initial state is uniformly sampled. $(S_F|s_0)$ indicates the next-state distribution given an initial state s_0, and n denotes the number of states. In simpler terms, EI quantifies the average difference in subsequent states given the constrained and unconstrained initial states of a system.

For a system with its microstates and dynamics, different microstate-macrostate mappings lead to different coarse-grained systems. Therefore, within the causal emergence theory, the identification of macrostates transforms into another task: finding an optimal microstate-macrostate mapping that maximizes the effective information. According to their definition, macrostates are identified when the past and future of different macrostates are distinguishable (as depicted in Figure 1B). Here, symmetry is about exchangeable in the mapping between past and future, leading to a conservation of distinguishable macrostates. However, it is essential to note that causal emergence is defined based on discontinuous dynamical systems evolving over time, and not all regularities we may wish to associate with laws will involve time. For example, text-image mapping (Rombach et al. 2022), genotype-phenotype mapping (Ahnert 2017), or parameter-pattern mappings (Gray and Scott 1984) are all complex systems with regularities yet to be uncovered. Consequently, causal emergence is not sufficiently general to allow for identifying laws in complex systems. A more general theory is needed.

1.5 Incidental Identification of Macrostates Through Alternative Methods

Considering the fundamental significance of macrostates, they are often implicitly identified – even in contexts where the term is not explicitly employed, or where the primary focus lies elsewhere. This incidental recognition is particularly notable within the domain of machine learning. Due to the high dimensionality of the data processed in machine learning tasks, dimensionality reduction techniques are routinely employed. Given the fact that many dimensionality reduction methods entail some degree of information loss, dimension reduction mapping often becomes a many-to-one mapping. This implies that distinct points in the phase space (i.e., different images, words, etc.) may be mapped to the same lower-dimensional representation. As per the general definition of macrostates stated earlier, these methods are intrinsically linked to macrostates. Noteworthy dimensionality reduction techniques include principal component analysis (PCA) (Jolliffe and Cadima 2016), t-SNE (Van der Maaten and Hinton 2008), auto-encoders (AEs) (Hinton and Salakhutdinov 2006), and variational auto-encoders (VAEs) (Kingma and Welling 2013; Higgins et al. 2017). Notably, the word embedding methods (Mikolov, Sutskever, et al. 2013; Mikolov, Chen, et al. 2013) and contrastive learning methods (He et al. 2020; T. Chen et al. 2020; Chen and He 2021) can be directly connected to the study of causal emergence. I will detail the word embedding and contrastive learning methods in the following sections.

1.5.1 Word Embedding

Word	**One-Hot Encoding**	**Index**
cat	$[1, 0, 0, 0, ...]$	1
dog	$[0, 1, 0, 0, ...]$	2
frog	$[0, 0, 1, 0, ...]$	3
...	$[0, 0, ..., 1, ...]$	...

Table 1. One-hot Encoding Example.

A particularly notable implementation is found in word embedding (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013). English words, despite typically consisting of a countable number of characters, lack inherent similarity. As such, people often represent words using high-dimensional one-hot encoding vectors, where only the n-th item is set to one, while all others remain zero (see Table 1). Here, n is merely the index of the word, bearing no inherent meaning as it can be interchanged without any loss of information. In one-hot encoding, any two words are orthogonal, indicating an absence of embedded similarity information.

Humans can effortlessly discern that “cat” and “dog” are similar as they are both animals. But how can a computer make such a connection using only indices? A simple non-trivial approach is to assign each word i a vector w_i. The cosine similarity between two word vectors then represents the similarity between the two corresponding words. The optimization of these word vectors typically employs a context-based approach: words appearing in similar contexts tend to exhibit similarity. Here, “context” means the words j surrounding the target word i. From a macrostate viewpoint, it is vital to note that context similarity is also defined by words: similar contexts will contain similar target words (the word surrounded by the context words). In training word vectors under the skip-gram framework, the loss function is defined as:

$$\mathcal{L} = \mathbb{E}_{(i,j)\sim P_{\text{data}}} \left(\log \sigma(w_i \cdot v_j) + \sum_{n\sim P_{\text{negative}}}^{k} [1 - \sigma(w_i \cdot v_n)] \right), \tag{1.3}$$

where σ denotes the sigmoid function, with $\sigma(x) = 1/(1 - e^{-x})$. P_{data} signifies the distribution of word-context word pairs, while w and v are two distinct matrices of shape (N, d), where N represents the total number of words, and d is the dimensionality of word vectors. It is important to note that w and v are not identical matrices (Nalisnick et al. 2016), which broadens the applicability of this framework to assign different representations to words and their context. The P_{negative} is employed for sampling negative pairs to avoid mode collapse.

Upon comparing this method to causal emergence theory, a key commonality emerges: both methods define the similarity of words (or microstates) based on the macrostate (w_i and v_j) of their related states, not the microstate (one-hot encoding of i and j) of these related states. This circular definition may lead to some seemingly arbitrary results. For instance, the distance function exhibits rotational symmetry, that is, $\mathcal{L}(w, v) = \mathcal{L}(Rw, Rv)$, where R represents any rotation matrix. Another aspect that might not be entirely constrained is the dimensionality. The embedded points could potentially lie on a lower-dimensional manifold within the high-dimensional space. However, this flexibility also enables the recognition of more generalized macrostates, which is beyond the scope of causal state theory.

1.5.2 Contrastive Learning

Contrastive learning originated in the computer vision domain and was originally formulated in a manner distinct from macrostate theory (He et al. 2020; T. Chen et al. 2020; Chen and He 2021). Conventional computer vision tasks typically function as classifiers, such as AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and ResNet (He et al. 2016): given an image, a label is predicted. In contrast, contrastive learning acts as a representation learner. When presented with one image, and another, it determines if the two images are identical under some transformation. For instance, if we present a colorful image of a cat and its gray-scale equivalent, a contrastive classifier will return “true” (these are known as positive pairs). However, if the second image were a dog or a gray-scale image of a different cat, we would expect it to return “false” (known as negative pairs). To achieve this objective, contrastive learning commonly utilizes an encoder f to map x_i to its corresponding representation z_i, subsequently leveraging the similarity of these representations to perform classification. A typical contrastive learning approach formulates the training loss for a positive pair (i, j) as follows (T. Chen et al. 2020):

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=[1,2N]\setminus\{i\}} \exp(\mathrm{sim}(z_i, z_k)/\tau)}, \tag{1.4}$$

where $\mathrm{sim}(z_i, z_j)$ refers to a similarity function for two representations z_i and z_j, often utilizing the cosine similarity. And τ is a temperature parameter that determines the contrastiveness of different similarity. Samll τ will enlarge small similarity differences. The k in the denominator denotes the negative pairs. In summary, this loss function aims to increase the similarity between i and j, while reducing the similarity between i and all other samples (negative samples).

This process enables contrastive learning to be trained to learn certain symmetries such as rotation, translation, and flipping, thus obtaining a lower dimensional representation without labeling. However, as these symmetries are all human-imposed, there is an unavoidable tendency to overlook some less evident symmetries that may not be immediately recognized by human observers.

Contrastive predictive learning (Oord, Li, and Vinyals 2018) mitigates this flaw and has demonstrated superior performance in numerous tasks, even when compared to supervised learning. Contrastive predictive learning utilizes the encoding of past states to predict the encoding of future states. Given the non-directional nature of this prediction, it could also be stated that it uses the encoding of future states to predict the encoding of past states. As the training pair is not generated by applying transformations, it is not constrained by human-imposed symmetry, thus permitting the learning of more complex symmetries. When compared to causal emergence theory and the word embedding method, contrastive predictive learning also shares a key commonality. All three utilize encoding, or as I would prefer to say, macrostates, to predict another macrostate rather than focusing on microscopic details. The advantage of contrastive predictive learning is its use of neural networks to represent the transition from microstates to macrostates. However, contrastive predictive learning requires a large number of negative samples to avoid trivial solutions, which results in substantial computational costs. Simultaneously, this method is incapable of sampling microstates given a particular macrostate, thereby restricting its applicability.

1.6 Integrative Perspective of Macrostate Theories

Upon critically reviewing the preceding theories and methods, recognizing their limitations, and identifying their commonalities, the concept of macrostates gradually becomes clear. We can adopt an integrative perspective to enhance our understanding of macrostates.

In essence, macrostates of systems can be regarded as invariant measures under specific mappings. A similar definition has also been proposed by (Gömöri, Gyenis, and Hofer-Szabó 2017) using set theory language. In this context, the term “mapping” broadly encompasses any quantity related to these systems. For example, in causal state theory, the mapping corresponds to a past-future state mapping; in word embedding, it is word-context mapping; and in contrastive learning, the mapping is the transformations.

The previously discussed theories and methods regarding macrostates can be broken down into two key components. And, by generalizing these elements, we can

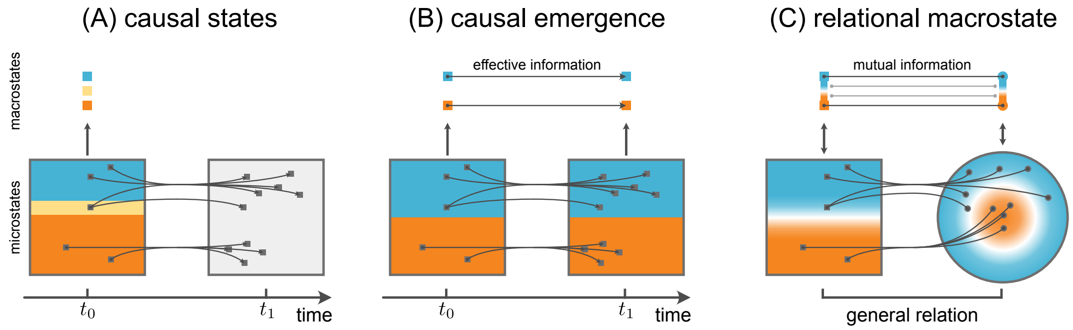


Figure 1. Comparison Between Causal State Theory, Causal Emergence Theory, and the Relational Macrostate Theory Presented in This Work. (A) in Causal State Theory, Two Microstates Are Equivalent (Belong to the Same Macrostate) If Their Future Microstate Distributions Are the Same. (B) Causal Emergence Theory Identifies Macrostates Where the Past-future Mappings Are Deterministic and Non-degenerate at the Macro Scale, Such That the Macrostates Can Be Distinguished from One Another in the past (Non-degeneracy) and the Future (Determinacy). Both Causal State Theory and Causal Emergence Theory Define Macrostates in Terms of Temporal Relations Within a System of Interest, as Denoted by the Square Shape Underlying the Mapping. (C) in the Relational Macrostate Theory We Propose Here, Two Microstates Are Equivalent If They Relate to the Same Macrostate Distributions, Which Can Be Generalized to Any Type of Relation, Including Past-future, Rule-pattern, Genotype-phenotype, Etc. – the Square and Disk Shapes Denote Generality by Visualizing How Macrostates Can Be Constructed as Maps That Exist Across Different Spaces.

develop a more comprehensive theory of macrostates. First, to define a macrostate, we must define the similarity (or, in extreme cases, the equivalence) of microstates, which reveals the similarity of their respective macrostates. The microstate-macrostate mapping can be then derived from this similarity. The second component involves choosing an appropriate mapping. For instance, theories such as causal state theory and contrastive predictive learning select past-future mapping.

From this viewpoint, a general macrostate theory naturally emerges: Firstly, I define the similarity of microstates in terms of their associated **macrostates**. Secondly, the mapping, or the relation, can be more general. Unbounded by temporal or human-imposed transformations, it could even be a mapping between two entirely different concepts, such as rule and pattern, or genotype and phenotype. Temporal relations are naturally encompassed as a special case that can yield invariant quantities.

In the following chapters, I will first introduce the relational macrostate theory, encompassing both the mathematical framework and a machine learning architecture, namely MacroNet, purposed for the identification of macrostates and the sampling of microstates possessing certain macrostates. I will apply the theory to several simulated systems, demonstrating its effectiveness and its capacity for learning meaningful macrostates. In the subsequent chapter, I will then apply the macrostate theory to a complex chemical system, namely oil-droplet systems, wherein oil droplets composed of different chemical compounds move within Petri dishes. Utilizing the macrostate theory and MacroNet, I will be able to identify macrostates within this system and subsequently answer the question: “Which combinations of parameters control which aspects of the oil droplet movement patterns?”

When comparing various domains such as physics, machine learning, information theory, and the domain of complex systems, it becomes evident that the concept of macrostate serves as a central unifying thread, succinctly linking these fields. As depicted in Figure 2, each domain has its own terminology or proxy concept that closely relates to the idea of macrostates. In physics, as previously discussed, macrostates are pivotal for formulating laws. Concepts like invariants can be viewed as specific types of macrostates. Furthermore, there’s an intrinsic link between symmetries and macrostates based on their fundamental definitions. In the machine learning sphere, what we term macrostates often go by names such as latent space or embeddings. These are typically identified through methods like contrastive learning. In the realm of information theory, mutual information is important in describing interactions between random variables. In the context of complex systems, macrostates find application

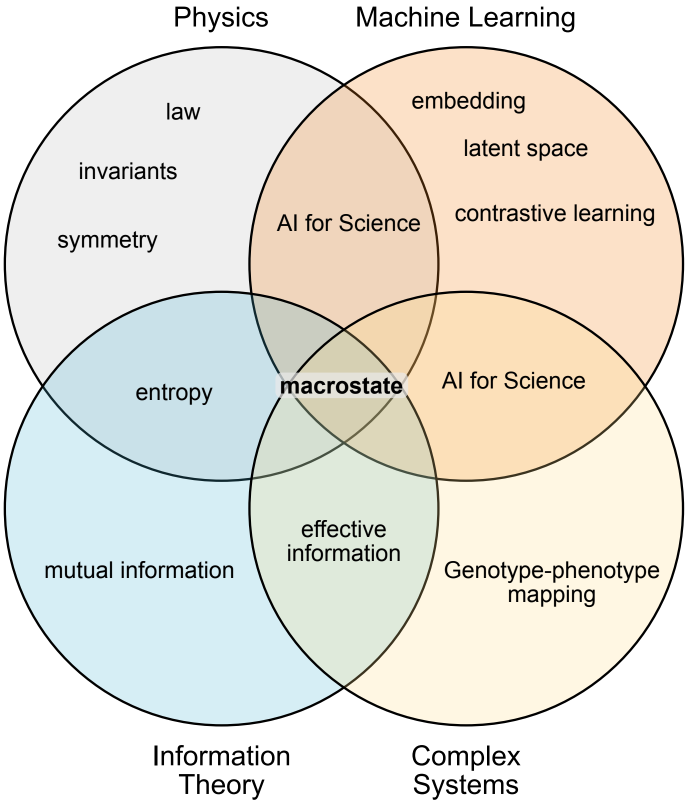


Figure 2. The Venn Diagram Illustrates the Relationship of Macrostates with Various Domains, Encompassing Physics, Machine Learning, Information Theory, and Complex Systems. The Review and Comparison of These Domains Highlight the Unique Role of Macrostates in Bridging These Diverse Topics.

in describing genotype-phenotype mappings, especially given their inherent many-to-many mapping characteristics. A contemporary and rising domain is AI for science, which leverages machine learning as a bridge to physics and complex systems. A primary goal in both AI for science and physics is the quest for discerning laws and regularities. Consequently, this domain, too, requires the identification of macrostates. One of the key insights the macrostate theory offers to the field of machine learning, particularly in the context of AI for science, is the imperative role of generative models in identifying regularities and laws. As discussed in section 1.3, our observations are combinations of laws and boundary conditions. Consequently, if our objective is to distill laws from these observations, we must be prepared to discard the information associated with boundary conditions. Delving deeper technically, our optimization goals shouldn't be focused on predicting these observations since we're intentionally omitting certain information. Instead, our approach needs to center on training generative models that predict distributions of observations, as they present a more fitting framework for uncovering these underlying laws. A noteworthy overlap between physics and information theory lies in the concept of entropy. While Shannon entropy and Boltzmann entropy appear distinct on the surface, deep interconnections have been identified (Jaynes 1957). In the context of physics, entropy requires a predefined macrostate for its determination. Although this is straightforward for systems like gases, the macrostates for more complicated systems remain ambiguous. Therefore, the entropy of these systems remains undefined in the absence of a clear macrostate. The theory of causal emergence, which melds information theory with complex systems, utilizes the concept of effective information, further underscoring the significance of macrostates. Drawing all these interconnections together, it becomes increasingly apparent that the macrostate stands as the pivotal concept, illuminating and bridging the gaps among these diverse fields, with the potential to significantly influence each of them.

Chapter 2

ARTIFICIAL NEURAL NETWORKS AND GENERATIVE MODELS

2.1 Abstract

When trying to identify macrostates, researchers often rely on observed data. This process typically involves mapping microstates to their corresponding macrostates, which necessitates the discovery of functions that meet certain criteria. Given the complexity of the systems under investigation, artificial neural networks are required. These networks, composed of simple linear and non-linear transformations, can be trained to adapt and improve. In this chapter, we will begin with an introduction to the foundational concepts of neural networks. Subsequently, we will delve into generative neural networks, including auto-encoders and normalization flows.

2.2 Introduction

While the fundamental definition of a macrostate is the subsets of microstates, in practice, the identification of a macrostate involves the discovery of a microstate-macrostate mapping. However, neither the causal state theory nor the causal emergence theory defines the macrostate via a microstate-macrostate mapping. The causal state theory frames a macrostate through the lens of equivalence, while word embedding and contrastive learning identify macrostates by establishing similarity. This necessitates a method to find a mapping function that fits the definitions. Artificial neural networks learn functions through the use of loss functions, without the need for manual design of the mapping function. As a result, it becomes the optimal choice for identifying macrostates.

Over the past few decades, particularly in the most recent one, artificial neural networks have yielded significant breakthroughs. They have addressed a plethora of crucial tasks by learning data distributions without necessitating manual detailing. These networks can extract generalizable knowledge from large data sets, even those with complex patterns. This trait positions neural networks as powerful tools for understanding complex systems. Indeed, the topic of "AI for Science" is getting increased attention, as neural networks can facilitate scientific research across numerous domains, such as Alpha Fold 2 (Jumper et al. 2021), AI Feynman (Udrescu and Tegmark 2020), the rediscovery of Schrödinger's equation (Wang, Zhai, and You 2019), among others. My subsequent works and studies, which involve empirical data and complex systems, also rely on artificial neural networks and generative models. Given their extensive technical details, and considering that my primary contribution does not lie within the realm of neural networks, I will introduce them in this separate chapter.

2.3 Artificial Neural Networks

Artificial neural networks can often be considered as a trainable function, $y = f(x)$, where x is the input, and y is the output. In most cases, neural networks are been trained to minimize loss functions, which is particularly important for finding macrostates via the definition of similarities. The key task is to parameterize the function f. Artificial neural networks often parameterize the function in a compositional way – combining simple units into a complex layered network. Each unit

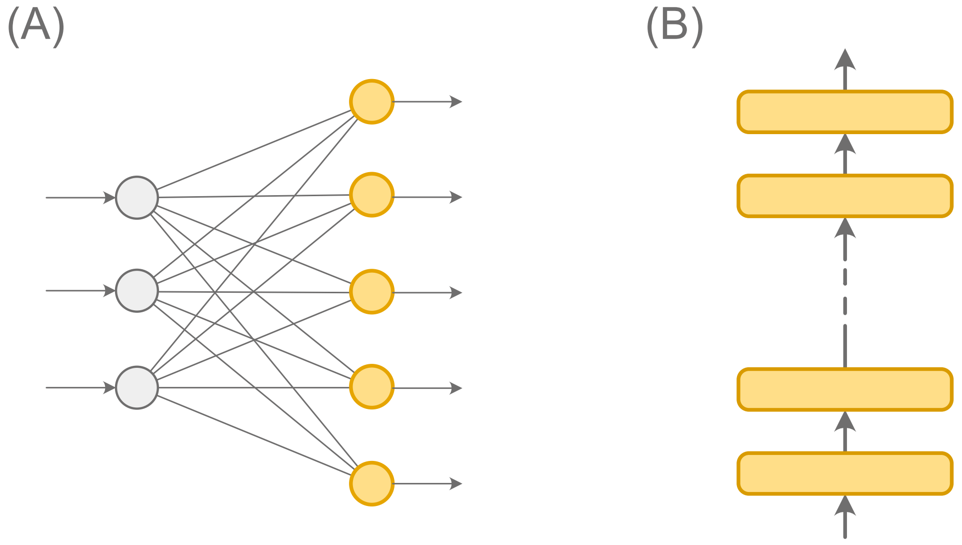


Figure 3. Typical Visual Representation for Neural Networks. (A) a Single-layer Neural Network Is Often Visualized in a Network-oriented Manner. The Bias and Nonlinear Terms Are Frequently Not Indicated. The Arrows Denote the Input and Output. (B) When Visualizing Deep Neural Networks, an Fcn Layer Is Typically Simplified into a Single Block.

often has trainable parameters that have been trained by gradient descent (Goodfellow, Bengio, and Courville 2016) (or other variations) algorithm. The most frequently used basic unit in neural networks is linear layers and activation functions (Goodfellow, Bengio, and Courville 2016).

A linear layer, mathematically is a $n \times m$ weight matrix $\mathbf{W}$ and a bias vector $\mathbf{b}$ with dimension n, where m will be the input dimension, and n will be the output dimension. In machine learning, the dimension has often called the number of features. Mathematically, a linear layer is defined as:

$$f(x) = \mathbf{W}x + \mathbf{b} \tag{2.1}$$

Combining multiple matrices by repeating doing matrix product will still give us a matrix. Hence, it will have no difference from linear regression with linear layers only. So, in order to represent or fit more complex functions, a nonlinear layer is required. Some typical nonlinear layers include sigmoid, ReLU, Tanh, and more. As an example, the sigmoid function is:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \tag{2.2}$$

Use g to represent the nonlinear layer, a typical fully-connected-network (FCN) can be formulated as:

$$f_{\text{FCN}}(x) = g(\mathbf{W}x + \mathbf{b}). \tag{2.3}$$

A single FCN layer often lacks the ability to perform complex computations. In fact, a single FCN layer, also referred to as a perceptron (Rosenblatt 1958), is incapable of fitting the XOR function. However, by stacking multiple linear and non-linear layers, we create deep neural networks—also known as deep learning models—that can solve complex problems by learning from data. Theoretically, a sufficiently large two-layer FCN can universally fit any continuous function (Cybenko 1989). Figure 3 showcases typical visualizations of neural networks, a style that I will utilize in subsequent chapters.

2.4 The Necessity of Generative Models for Macrostate Identification

As discussed in section 1.3, the quest to uncover physical laws necessitates the distinction of boundary condition information from observations. Similarly, macrostates, as critical components of physical laws, require such differentiation. Invariant quantities, as specific instances of macrostates, inherently comprise physical laws. For example, the energy conservation law in a simple harmonic oscillator, represented by $E = \frac{1}{2}mv^2 + \frac{1}{2}kx^2$, provides a motion regulation for v and x. Reflecting on the need for physical laws to be distinct from boundary condition information – i.e., initial states, environments, or noise – exact prediction becomes untenable. To illustrate, predicting a car's trajectory without knowledge of its starting point, date, or weather conditions is impractical. Indeed, an attempt to minimize the mean square error (MSE) for a prediction task will only yield an average trajectory, which might nonsensically place the vehicle at the Earth's core. Evidence for this is evident in image-to-image translation tasks – while simple predictions for translating labels into images yield only blurred images, generative models have the capability to sample clear images (Isola et al. 2017). However, recognizing it as a car retains some constraining information regarding its possible locations: it cannot be underwater, and it is most likely to be on a road. Essentially, the information retained after excluding boundary conditions encompasses the core behavior of a car, which differentiates it from a person or an airplane.

Since minimizing mean square error in prediction cannot extract and utilize this information, the need for generative models arises. In a conditional generative scenario, the possible trajectories can still be sampled without knowing the car's starting location. The car could follow trajectories on any road, but not any river or sea. Despite the randomness in sampling, the model captures the key differences between a car and other objects, such as a person. For instance, when sampling the position of a person, it will assign higher probabilities to buildings. As the previous example illustrates, a simple average prediction can lose all this information, making it harder to distinguish between a car and a person. In essence, attempting to predict the unpredictable may cause us to lose predictable information, an issue that conditional generative models can avoid. I will illustrate this effect in chapters 3 and 4.

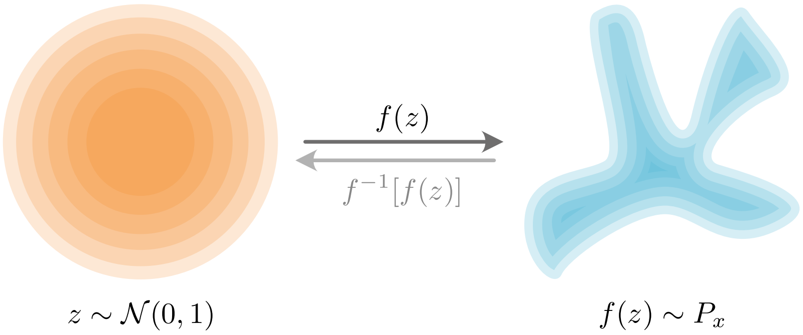


Figure 4. Generative Models Typically Employ Functions Applied to Simple Distributions, Such as Normal Distributions, and Train These Functions Such That the Transformed Distribution Aligns with the Data Distribution.

2.5 Generative Models

From a mathematical perspective, generative models are trained to learn a distribution P that minimizes its Kullback-Leibler (K-L) divergence with respect to the data distribution P_x. In practice, as most neural networks are deterministic, the source of randomness often originates from the input. For example, Generative Adversarial Networks (GANs) commonly employ inputs sampled from a high-dimensional standard normal distribution (Goodfellow et al. 2014). This approach is so prevalent that the main focus of current generative model research is to ascertain how a simple distribution can be transformed into a more complex one. Mathematically, given a random variable z that follows a simple distribution—such as a standard normal distribution—we seek to find a function f that transforms this simple distribution to fit the data, i.e., $f(z) \sim P_x$, see Figure 4. In the following sections, I will introduce autoencoder and normalizing flow models. Although GANs are a significant area of research, I will not be discussing them as they are not used in my current work.

2.5.1 Auto-encoder

A straightforward method for learning such distribution transformation functions is by learning an inverse function. This general concept leads to both variational auto-encoders (VAEs) and normalizing flow (refer to section 2.5.2). For auto-encoders (Hinton and Salakhutdinov 2006), two neural networks are typically trained as an encoder, denoted as f_E, and a decoder, represented as f_D. The computation flow can be represented as:

$$x \xrightarrow{f_E(x)} z \xrightarrow{f_D(z)} \hat{x} \tag{2.4}$$

The objective of auto-encoder training is to reconstruct the input:

$$\min_{\theta} \|x - f_D[f_E(x)]\|. \tag{2.5}$$

However, this transformation does not guarantee that the latent representation, $z = f_E(x)$, adheres to a simple distribution. As a result, the generative capacity of an auto-encoder is limited. To overcome this, the VAE has been proposed (Kingma and Welling 2013; Higgins et al. 2017). Unlike the traditional auto-encoder, a VAE learns the latent representation as a distribution. The distribution $P(z|x)$ is parameterized as μ and $\log \sigma$ by approximating it with a normal distribution. In addition to the reconstruction objective, the VAE also trains $P(z|x)$ to follow a standard normal distribution by minimizing the Kullback-Leibler (K-L) divergence between $P(z|x)$ and the normal distribution. In practice, the encoder computes μ and $\log \sigma$, then $z = \mu + \sigma w, w \sim \mathcal{N}(0, 1)$ is sampled as the input of the decoder. This process is termed reparameterization. Since z is trained to follow the standard normal distribution, the decoder f_D can learn the transformation from the normal distribution to the data distribution P_x. The computation flow of VAE is represented as the following diagram:

$$x \xrightarrow{f_E(x)} \mu, \sigma \xrightarrow{\mu+\sigma w, w\sim\mathcal{N}(0,1)} z \xrightarrow{f_D(z)} \hat{x} \tag{2.6}$$

However, even though $P(z|x)$ is trained to follow a normal distribution, $P(\mu|x)$ does not necessarily follow the same distribution. Following the proposal of VAE, InfoVAE (Zhao, Song, and Ermon 2017) was developed to enhance VAE, ensuring that the latent space is deterministic and follows a normal distribution without reparameterization. InfoVAE introduces a new loss term referred to as the Maximum Mean Discrepancy (MMD) loss. For a given latent distribution P, and a target distribution Q, MMD is calculated as:

$$\text{MMD}(P|Q) = \mathbb{E}_{P(z),P(z')}k(z,z') + \mathbb{E}_{Q(z),Q(z')}k(z,z') - 2\mathbb{E}_{P(z),Q(z')}k(z,z'), \tag{2.7}$$

where $k(z,z') = \exp(-\|z-z'\|^2/(2\sigma^2))$, and $/mathbbE$ represents the mathematical expectation. In comparison to VAE, InfoVAE deterministically encodes a latent representation z that follows a normal distribution. This makes InfoVAE an effective tool for data compression. In reality, most high-dimensional data, such as images, often represent lower-dimensional structures. A well-suited encoder can reduce its dimensionality while preserving essential information.

2.5.2 Normalizing Flows

Pursuing the same objective of learning transformations from simple to complex distributions, normalizing flow has been proposed as an alternative approach. normalizing flow introduces specially designed neural networks capable of straightforward

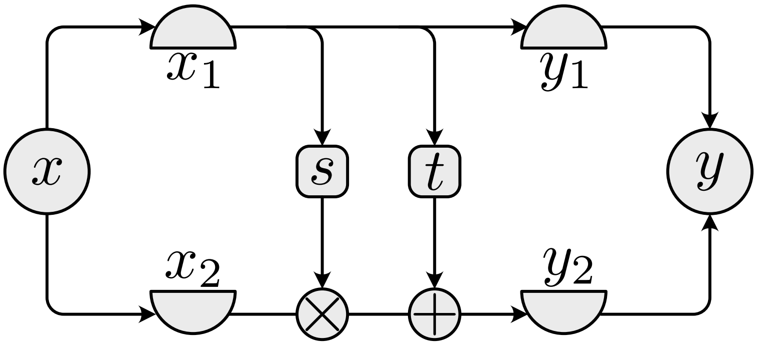


Figure 5. Real NVP Partitions the Input into Two Components, x_1 and x_2. By Preserving the Information of x_1, It Guarantees Its Invertibility and Simplifies the Computation of the Log Determinant of the Jacobian.

inverse computations, specifically Invertible Neural Networks (INNs). Given an input x, an invertible neural network f can readily perform the inverse operation such that $f^{-1}(f(x)) = x$. Several models have been proposed to ensure this invertibility, including NICE (Dinh, Krueger, and Bengio 2014), Real NVP (Dinh, Sohl-Dickstein, and Bengio 2016), ResFlow (R. T. Chen et al. 2019), Glow (Kingma and Dhariwal 2018), among others. For instance, and also for the purpose of our subsequent discussion, Real NVP has been specifically designed to guarantee invertibility. For an input $x_{1:n}$ with n features, the output $y_{1:n}$ is a concatenation of two parts, as shown in Figure 5:

$$y_1 = x_1 \tag{2.8}$$

$$y_2 = x_2 \exp(s(x_1)) + t(x_1), \tag{2.9}$$

where $x_1 = x_{1:m}$, $x_2 = x_{m+1:n}$. The inverse function is easily obtained:

$$x_1 = y_1 \tag{2.10}$$

$$x_2 = [y_2 - t(y_1)]/\exp(s(y_1)). \tag{2.11}$$

As y_1 is equal to x_1, a swapping layer that swaps x_1 and x_2 is often used in practice to ensure all features are computed. Other works like NICE, or ResFlow, have specifically designed networks with constraints to maintain this invertibility, see Appendix A.2.2.

Once the inverse problem is addressed, the focus shifts to training the output to follow a normal distribution. Given an INN f, for which $f(x) \sim \mathcal{N}^n(0, 1)$, sampling is made easy by performing an inverse pass $f^{-1}(z), z \sim \mathcal{N}^n(0, 1)$. To generate samples that follow the data distribution, normalizing flow directly maximizes the log probability density $P_X(x = f^{-1}(z))$ of generated samples. Utilizing the change of variable formula, the probability is given as:

$$P_X(x) = P_Z(z) \left| \det \frac{\partial f^{-1}(z)}{\partial z^\top} \right|^{-1}, \tag{2.12}$$

where P_Z is the prior distribution of z, and $\det \partial f^{-1}(z)/\partial z^\top$ is the determinant of the Jacobian. By maximizing the expectation of $\log P_X(x)$, the generated samples will follow the data distribution. Since f is invertible, the encoding of $x \sim P_X$ will also follow the prior distribution. In practice, the log probability is computed as:

$$\log P_X(x) = \log P_Z(f(x)) + \log \left(\left| \det \frac{\partial f(x)}{\partial x^\top} \right| \right). \tag{2.13}$$

Since the log determinant of the Jacobian is needed, normalizing flow models must also be able to compute this quantity easily. For a general neural network, computing such a quantity requires substantial computational power given the Jacobian matrix is $n \times n$ shaped. However, normalizing flow models utilize their special design again to make such computations more feasible. For example, the Jacobian matrix of Real NVP is:

$$\frac{\partial f(x)}{\partial x^\top} = \begin{bmatrix} \dfrac{\partial y_{1:m}}{\partial x_{1:m}^\top} & \dfrac{\partial y_{1:m}}{\partial x_{m+1:n}^\top} \\ \dfrac{\partial y_{m+1:n}}{\partial x_{1:m}^\top} & \dfrac{\partial y_{m+1:n}}{\partial x_{m+1:n}^\top} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \dfrac{\partial y_{m+1:n}}{\partial x_{1:m}^\top} & \mathrm{diag}\left(e^{s(x_{1:m})}\right) \end{bmatrix} \tag{2.14}$$

Because this Jacobian matrix includes a zero term on the top-right corner, its log determinant is simply the sum of the elements of $s(x_{1:m})$, and we don't need to compute the complex term of $\partial y_{m+1:n}/\partial x_{1:m}^\top$. While other normalizing flow models may use different methods, the primary goal is always to compute or estimate the log determinant of the Jacobian in a cost-effective manner.

Due to the requisite of invertibility, all normalizing flow models' output dimensions match the input dimensions. This characteristic can make using normalizing flow as an encoder challenging, as encoders often necessitate an output with a significantly lower dimension than the input. However, in the study of macrostates, dimension reduction becomes critical because we want to find lower dimensional macrostates. An approach is to simply disregard certain dimensions inspired by (Hu et al. 2022). Given a normalizing flow f, with an output $A = f(x)$, the dimension-reduced vector is $\alpha = A_{1:k}$, where k is the desired dimension. Subsequently, only α is used for downstream tasks. When inverting the function, given the encoding α, the output is $f^{-1}([\alpha, z])$, where z is sampled from the prior distribution, and $[\alpha, z]$ means concatenate two vectors. For this reason, despite some dimensions being disregarded, they are still trained to follow the prior distribution. This process automatically allows us to perform conditional sampling $P(x|f(x)_{1:k} = \alpha)$. This method will be used in the following chapters for conditional sampling and parameter design.

Normalizing flow models, being capable of acting as encoders, controlling the latent space distribution, and performing conditional sampling based on the latent space, serve as unique tools for identifying macrostates and designing microstates. First, macrostates need to be encoded from microstates; second, the macrostate distribution needs to be controlled to avoid trivial solutions, such as constant values and lower-dimensional manifolds; and third, to sample microstates, conditional sampling based on the latent space should be implemented.

It's important to note that Variational Autoencoders (VAEs) cannot meet all three of these requirements simultaneously. Although VAE or InfoVAE can control their output distribution, they cannot perform conditional sampling based on their latent spaces, as the inherent randomness originates from these latent spaces. More importantly, the primary training objective of an autoencoder is to reconstruct the input, meaning similar encoding will result in similar output. From the perspective of symmetry, this only captures the symmetry of local small changes, while ignoring translation, rotation, and other more complex symmetries. This topic will be discussed in greater detail in Chapter 3.

Chapter 3

RELATIONAL MACROSTATE THEORY GUIDES ARTIFICIAL INTELLIGENCE TO LEARN MACRO AND DESIGN MICRO

3.1 Abstract

The burgeoning field of AI for Science has emerged as a pivotal domain. Creating systems that display specific behaviors, often referred to as “parameter design,” is now a key focus in this area. A key challenge in parameter design is the classification problem, which involves properly categorizing objectives for the purpose of designing appropriate parameters, particularly when the parameter-pattern mapping is stochastic. This challenge is closely linked to the concepts of symmetry and macrostates, where macrostates provide a unifying framework for understanding symmetry and classification. In this paper, we propose a novel relational macrostate theory that defines macrostates based on the relationship between microstates of patterns and parameters, offering a fresh perspective on macrostate identification. Furthermore, we introduce MacroNet, a neural network capable of identifying macrostates and designing parameters in a contrastive generative manner. Our method demonstrates remarkable success when applied to Turing patterns, a complex dynamical model for studying pattern formation in biology, showcasing its potential to contribute significantly to the AI for Science domain.

3.2 Introduction

The burgeoning field of artificial intelligence (AI) has increasingly contributed to scientific inquiry in recent years, with rapid advancements in AI algorithms and technologies. Notable examples in this domain include AI Feynman (Udrescu and Tegmark 2020), which discovers physical laws from observation; the application of machine learning to uncover hidden state variables (B. Chen et al. 2021); and AlphaFold 2, which employs machine learning to predict protein folding (Jumper et al. 2021). One central aspect of the "AI for Science" domain is parameter design, particularly for complex systems. This involves creating rules or parameters for generating patterns, such as designing genes for producing specific proteins or finding parameters for generating desired patterns. The genotype-phenotype mapping problem is crucial in various scientific disciplines and has garnered significant attention.

A fundamental challenge in parameter design is the classification problem, which entails proper categorizing objectives for the sake of designing appropriate parameters, especially when the parameter-pattern mapping is stochastic due to thermal noise or initial states. For instance, given an example pattern, researchers aim to identify parameters that can generate similar patterns, necessitating a definition of similarity. Ultimately, this task requires defining symmetry, as it pertains to the equivalence of various patterns. Imposing symmetry is a vital aspect of the AI for Science domain, with notable examples including AlphaFold 2, which employs an equivariant neural network to impose rotational symmetry (Jumper et al. 2021), and classical neural networks such as convolutional neural networks, which impose translational symmetry.

From a physical perspective, the classification problem and the concept of symmetry are closely related to the physics concept of macrostates. Macrostates can be regarded as a continuous version of classification labels, with each macrostate associated with an ensemble of microstates. In terms of classification, microstates are grouped into the same class. Regarding symmetry, a macrostate possesses its intrinsic symmetry, such as the macrostate “apple,” which includes rotational, reflective, and various complex symmetries, allowing it to contain many apples with different microstates. An essential aspect of this connection between symmetry and classification is the role of macrostates in the broader scientific context. The abstraction of macrostates enables the study of general laws and principles, moving away from a case-by-case analysis, which is critical for advancing scientific understanding. This further highlights the significance of macrostates as a fundamental element of AI for Science, particularly in parameter design. However, many symmetries may be more complex and difficult to identify. Consequently, macrostate identification poses a challenge, especially for complex systems, warranting the use of machine learning to address this issue.

Traditional classification and parameter design objectives often evoke the idea of encoder-decoder architectures, such as variational autoencoders (VAEs) (Kingma and Welling 2013; Higgins et al. 2017) or models built upon VAEs. However, these models are trained with the objective of reconstructing microstates. In other words, encoder-decoder-based models define similarity based on the similarity of microstates, which can ignore many important symmetries, such as rotational symmetry, translational symmetry, and various complex symmetries. This ignorance overlooks the most crucial aspects of science, extracting only lower-dimensional representations for low-dimensional manifolds embedded in high-dimensional spaces (such as images). If the task focuses solely on classification and finding symmetries, we can adopt contrastive learning, which essentially trains neural networks to encode equivalent inputs into the same representation. However, contrastive learning methods often require a vast amount of negative samplings and do not natively support generative sampling ability, while parameter design, a critical objective, inherently involves generative or sampling processes. So, in this paper, we propose a novel relational macrostate theory that defines symmetry based on the relationship between patterns and parameters, adopting a relationalism perspective. We also introduce a neural network capable of identifying macrostates and designing parameters in a contrastive generative manner. We apply our method to Turing patterns (Pearson 1993; Gray and Scott 1984), a complex dynamical model for studying pattern formation in biology, demonstrating its effectiveness in classifying these patterns and sampling ensembles of parameters capable of generating patterns within the same macrostate.

3.3 The Relational Macrostate Theory

Macrostates and microstates are fundamental concepts in the study of complex systems, providing a framework for understanding the behavior and properties of these systems. Macrostates can be regarded as classification labels for microstates, which can be either discontinuous, as with conventional labels, or continuous, such as dimension reduction. In essence, macrostates emerge through a process of coarse-graining or information reduction, where a state is considered a microstate before coarse-graining and a macrostate after this process.

The need for a relational macrostate theory arises from the limitations of defining equivalence and similarity based solely on microstates, as discussed in the introduction. For example, consider a simple harmonic oscillator with microstates $u_1 = (p, x)$ and $u_2 = -u_1$. These microstates are distinctly different, but they possess the same energy, which is a vital macrostate property. Imposing rotational symmetry could address this issue; however, this requires domain knowledge, which is not suitable for a data-driven paradigm. To avoid overlooking important macrostates by focusing on microstate details, it is essential to consider the relationships between states, particularly for paired microstate data such as parameter-pattern or genotype-phenotype relationships. There have been several efforts focused on identifying macrostates associated with the emergent regularities found in complex systems (Udrescu and Tegmark 2020; Liu and Tegmark 2021). Despite the advancement in these theories or methods, they continue to rely upon microstates (B. Chen et al. 2021) or the distribution of microstates (Shalizi and Moore 2003) to delineate macrostates. Certain theories can encapsulate symmetries transcending micro-similarities, yet their applicability remains confined to time-series data (Hoel 2017). In prior research, a set-theory-based definition of macrostate has been proposed (Gömöri, Gyenis, and Hofer-Szabó 2017). However, the absence of elements from probability and information theory inhibits its comprehensive application to empirical data and restricts the use of recently developed methods. In the relational macrostate theory, we begin by implicitly defining equivalence between microstates. Consider two microstates $u \in U$ and $v \in V$ as two random variables. Their micro-to-micro relation can be mathematically represented as a joint distribution P(u,v). The u and v can be mapped to macrostates α and β respectively by φ_u and φ_v. So, we can also define a micro-to-macro relation by the joint distribution $P(\alpha, v)$ and $P(u, \beta)$. For a given microstate u_i (or v_i), its micro-to-macro relation can be represented as a conditional distribution $P(\beta|u_i)$ (or $P(\alpha|v_i)$). Then, we can define macrostates in the most (relational) general case as:

Definition 1. Two pairs of microstates u_i and u_j (and v_i and v_j) belong to the same macrostate if and only if they have the same micro-to-macro relation:

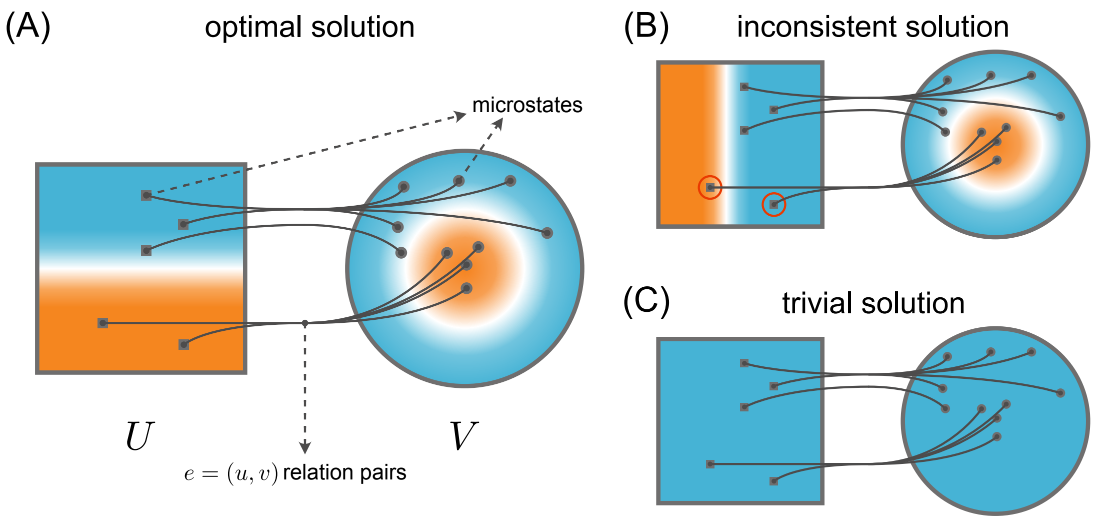


Figure 6. Macrostates Are Determined by Symmetries That Define Relations Between Ensembles of Microstates. The Rectangle and Disks Represent the Space of Microstates. And the Points and Links Represent the Observed Microstate Pairs (u_i, v_i). The Background Color of the Points Illustrates Their Macrostates. (A) An Optimal Solution. (B) An Inconsistent Solution. (C) A Trivial but Legal Solution, Which Coarse Grains All Microstates to the Same Macrostate. This Kind of Coarse Graining Is Not Informative Since the Mutual Information of Macrostates Is Zero, We Add an Information Theoretic Criterion to Identify Good Macrostates and Exclude Such Cases.

$$u_i \sim u_j \iff P(\beta|u_i) = P(\beta|u_j) \text{ and} \tag{3.1}$$

$$v_i \sim v_j \iff P(\alpha|v_i) = P(\alpha|v_j) \tag{3.2}$$

Note, this defines an equivalence class of symmetries where $u_i \sim u_j$ and $v_i \sim v_j$ (where $\sim$ indicates "is equivalent to" under the symmetry operation).

To facilitate the implementation of machine learning techniques, we introduce an explicit definition of macrostates. The relational macrostate theory defines a macrostate as a vector that contains information discernible from both sides of the paired microstate data. This involves seeking two coarse-graining functions, $\varphi_u(u)$ and

$\varphi_v(v)$, such that $\varphi_u(u) = \varphi_v(v)$ (see Figure 6A and Figure 6B). This definition aims to capture the shared information between the two sides of paired microstate data, providing a more meaningful representation of macrostates. Furthermore, the explicit definition serves as a practical solution to the implicit definition, making it a suitable choice for implementation in practice. While the relational macrostate theory provides a valuable framework for understanding macrostates, it is important to recognize that some solutions may be trivial or uninformative, such as the case where all microstates are coarse-grained to the same macrostate (Figure 6C). To address this issue, an information theoretic criterion can be employed to identify good macrostates and exclude uninformative cases. Specifically, we aim to maximize the mutual information between $\alpha = \varphi_u(u)$ and $\beta = \varphi_v(v)$. When a trivial solution is found, the mutual information $I(\alpha, \beta)$ will be very low, or even zero. Employing this information criterion helps avoid such cases, ensuring a more meaningful representation of macrostates.

3.4 MacroNet: A Machine Learning Framework for Identifying Macrostates and Design Microstates

In the above formalization, a macrostate in U is defined by macrostates in V (i.e., macrostates are defined only in terms of their relations to other macrostates). This relational definition necessitates that we optimize the macrostate mapping iteratively to find an optimal solution. Thus, to implement the relational macrostates theory, we propose a self-supervised generative model for finding macrostates from observations (Figure 7A).

Our definition of macrostates can be satisfied by optimizing a microstate-macrostate mapping to predict other macrostates. Here we use φ_u and φ_v to represent the mapping

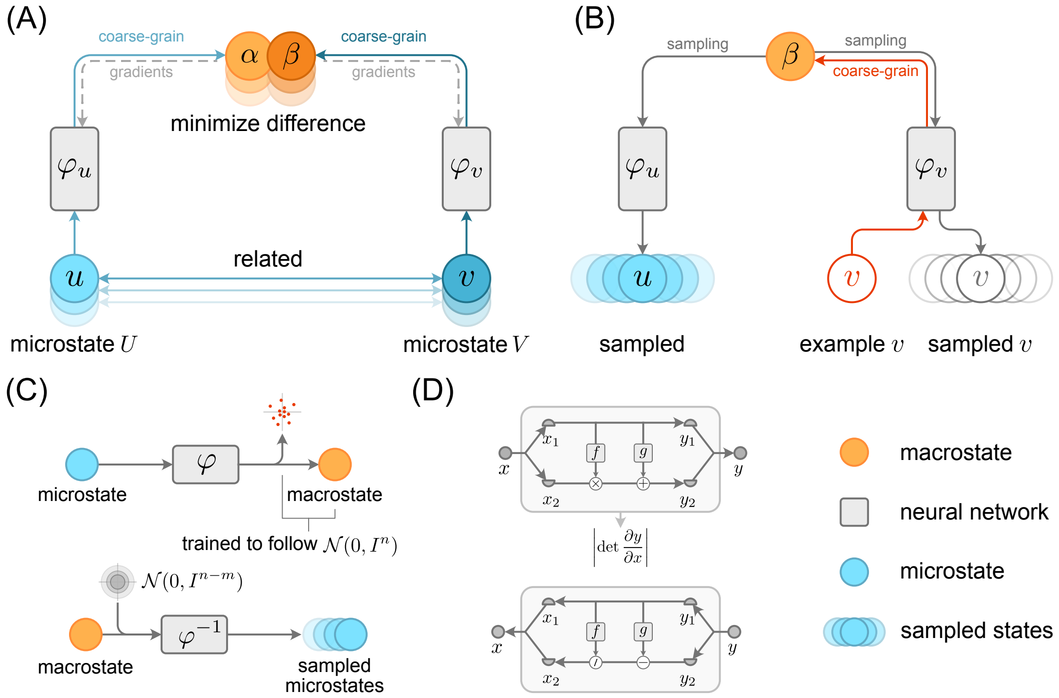


Figure 7. Neural Network Architecture of MacroNet. (A) During the Training Process, the Two Invertible Neural Networks Are Optimized to Map Two Types of Microstates to the Same Macrostates. These Microstate Pairs Can Correspond to Past and Future States, Dynamical Rules or Parameters and Observed Behavior, or Any Other Pair of (Sets of) Variables. (B) The Conditional Sampling and Designing Process. First, We Manually Make an Example Microstate of Type V and Compute Its Macrostate. We Then Sample the Microstates in U or V That Have This Macrostate. (C) When Doing Coarse-Graining, Parts of the Output Are Abandoned to Reduce the Dimensionality. The Abandoned Variables Are Still Trained to Follow an Independent Standard Normal Distribution. This Independence Makes It Possible to Do Conditional Sampling. (D) A Typical Invertible Neural Network Is RealNVP, Which Has a Specially Designed Structure That Guarantees Invertibility. The Log-Determinate of the Jacobian Is Also Easy to Compute for This Type of Neural Network for Controlling the Distribution of Their Outputs.

performed by the neural networks on U and V respectively. We have the prediction loss:

$$\mathcal{L}_P = \mathbb{E}_{(u,v)\sim P(u,v)}|\varphi_u(u) - \varphi_v(v)|^2, \tag{3.3}$$

where (u,v) are pairs of microstates sampled from the training data. The ideal solution for φ is $\varphi_u(u) \approx_\sigma \varphi_v(v)$, meaning the macrostate of u can be predicted by the macrostate of v with error of σ, and vice versa. However, we need an additional term to avoid trivial solutions such as a low dimensional manifold or constant. To do this, we add a distribution loss adopted from normalization flow models (Dinh, Sohl-Dickstein, and Bengio 2016), $\mathcal{L}_D = \mathcal{L}_{D_u} + \mathcal{L}_{D_v}$, where:

$$\mathcal{L}_{D_u} = \log P_{\text{normal}}(\varphi_u(u)) - \log \left|\det \frac{\partial \varphi_u(u)}{\partial u}\right| \tag{3.4}$$

$$\mathcal{L}_{D_v} = \log P_{\text{normal}}(\varphi_v(v)) - \log \left|\det \frac{\partial \varphi_v(v)}{\partial v}\right| \tag{3.5}$$

The distribution loss is minimized when the outputs follow independent standard normal distributions. We train the neural networks by combining the two loss functions with the hyperparameter γ, which is selected by experiments, often set to 1:

$$\mathcal{L} = \mathcal{L}_P + \gamma \mathcal{L}_D \tag{3.6}$$

Combining these two terms, we can approach the mutual information criterion. Directly computing $\mathcal{L}_D$ can be very expensive since it requires computing the Jacobian. However, since we want to do sampling, invertible neural networks (INNs, also known as normalizing flows) can help (Figure 7D). The INNs are not only designed to be invertible, but also designed to easily compute the log-determinant of the Jacobian. The INNs will have the same output dimension as the input, so we abandon part of the dimensions (Figure 7C). For example, if we want to map an 8-dimensional vector to two-dimensional macrostate, the INNs will still give an 8-dimensional vector as a result, but we only take the first two variables as the macrostate for training. The abandoned six variables, however, still have been trained to follow independent normal distributions so we can do conditional inverse sampling.

Given an example microstate v', suppose we want to find other microstates in V space with the same macrostate as v'. We can use $\varphi_v(v')$ to compute the macrostate β of the example v'. Then, we can invert the neural network to sample microstates v_s that have the same macrostates. This conditional sampling allows identifying the symmetry of macrostates and enables the design of microstates by sampling from a given target macrostate once the network is trained on other examples with the same macro behavior (Figure 7B). This kind of sampling can enable the design of complex systems: the identified macrostates are not given by humans, but instead, computed from examples by neural networks. This process makes it possible to design complex systems without needing to first classify their behavior.

3.5 Results

In what follows we consider three applications of MacroNet. We first demonstrate the key features of our workflow on linear dynamical systems, which allows easily demonstrating key concepts, via the identification of a rotational symmetry and design of microstates consistent with this behavior. The second example is a simple harmonic oscillator (SHO), where we demonstrate MacroNet can identify a familiar symmetry and its corresponding macrostate in physics – time translation invariance and energy. For this example, we show how our workflow can identify equal energy surfaces for the

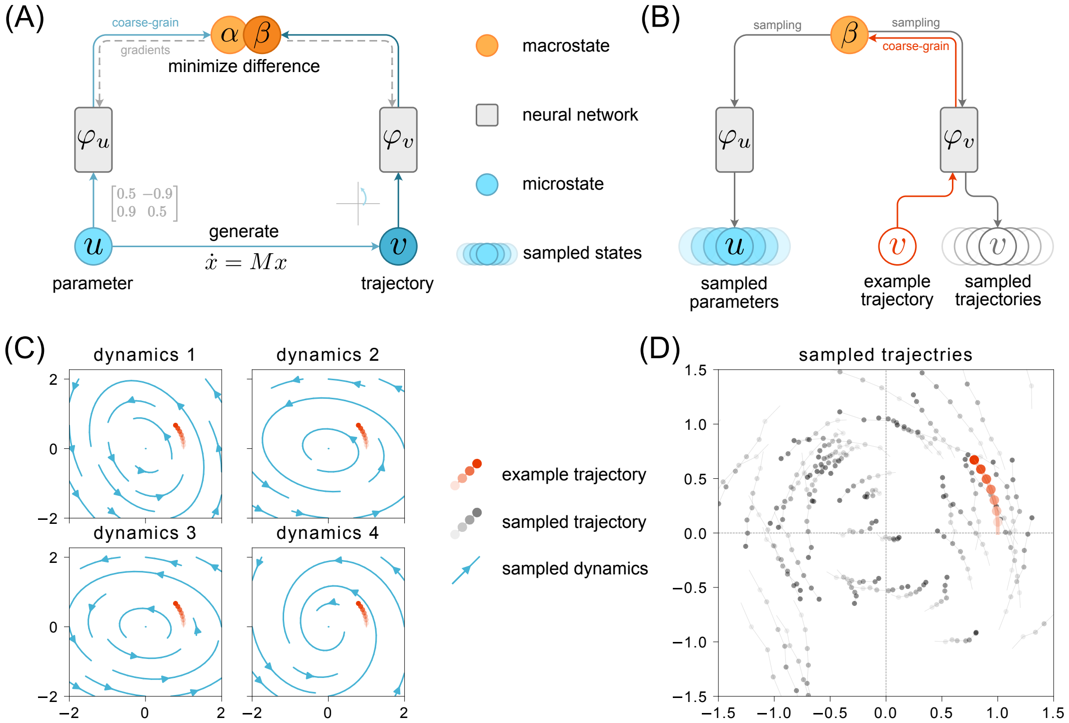


Figure 8. Training Neural Networks to Find Macrostates of Linear Dynamical Systems. (A) the Related Macrostate Pairs Are Distributions of Microstate Parameters and Trajectories. (B) by Choosing an Example Trajectory, We Can Sample Microstates of Parameters or Trajectories with the Same Macrostate. (C) given the Example Trajectory (Red Dots), We Can Compute Its Macrostate. Then, We Can Sample Other Sets of Parameters That Have the Same Macrostate. Using the Sampled Parameters, We Can Plot the Trajectories Generated by the Sampled Parameters (Blue Lines). (D) Using the Same Macrostate, We Can Sample an Ensemble of Trajectories That Have the Same Macrostate.

SHO. The final example is Turing patterns, where we show the utility of MacroNet in solving the inverse problem of mapping macro-to-micro in a complex system.

3.5.1 Linear Dynamical Systems

We start with an experiment analyzing linear dynamical systems. Their parameter-pattern mapping is a many-to-many relationship, which highlights the need for our methods. This experiment demonstrates the workflow of identifying macrostates based on symmetries and then designing microstates from the identified macrostates. Here we choose a two-dimensional linear dynamical system whose dynamics are given by

$$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = M\vec{x} \tag{3.7}$$

where x is the independent variable, and M is a 2×2 matrix that includes the parameters that specify the dynamics of the system. Given a matrix M and an initial state x_0, we can generate a sequence of observed states by computing $x_{t+1} = x_t + Mx_t\delta t$. The trajectory will be $T = [x_1, x_2, ..., x_n]$ in the two-dimensional space, where $n = 8$ and $\delta t = 1/n$. Here we choose n=8 because it is large enough to show the pattern of trajectories, but not so large as to slow the training. In this example, the micro-to-micro relations are represented by parameter-trajectory pairs, i.e., $(u, v) = (M, T)$.

The mapping between parameter and trajectories is a many-to-many mapping, which means: 1) given one parameter, different initial states will lead to different trajectories. 2) sampling different parameters may lead to the same or similar trajectories. We use two neural networks to learn the macro relation between parameters and trajectories: one uses φ_u to map the 4-d parameter matrix to a 2-d macrostate, and the other uses φ_v to map the 16-d trajectory to a 2-d macrostate (Figure 8A), where we optimize to maximize the mutual information between the identified macrostates in both cases.

After training, we can use the learned macrostates, which represent the conserved symmetries in parameter trajectory pairs, to design microstates. In Figure 8B, Given an example trajectory T_e, we can compute its macrostate $\beta = \varphi_v(T_e)$. The neural network φ_u^{-1} samples parameters that can generate trajectories for the example microstate (Figure 8C). The sampled parameters follow a conditional distribution $P(M|\beta)$, where M is the parameter matrix. In Figure 8C, we show how, given an anti-clockwise rotating trajectory, the parameters sampled all lead to anti-clockwise trajectories. Their macrostate embeddings are shown in SI, see Figure 26. By this process, we can design parameters of a system to mimic the behavior of any example, even without needing to translate the language describing the behavior to be human-interpretable. This ability has broad applicability for the design and control of complex systems, where simple mathematical descriptions have defied human scientists. Even when we do not know or have access to how we could describe a behavior, the neural network can still sample parameters to allow design of new examples through self-supervised learning.

So far, we have demonstrated sampling parameters for the matrix M, based on a specified macrostate (rotating anti-clockwise). We showed how the sampled parameters allow constructing new example trajectories using the sampled matrix M in Eq 3.2 with the desired macro behavior. We can also sample trajectories directly, via a sampling process where we specify the target macrostate and then use the inverse sampling to recover trajectories. These sampled trajectories follow the distribution of $P(T|\beta)$, where T is the trajectory microstate. Figure 8D show that the sampled trajectories all follow the same behavior, exhibiting anti-clockwise rotation, just as with the example trajectory. It is worth noting that we did not give the neural network any concept of "rotate" or "clockwise": the neural network discovered this symmetry on its own, as one that is relevant to how the parameters of the matrix M map to observed trajectories. This experiment gives a simple example of how a neural network architecture like MacroNet could potentially be developed that might aid in identifying more complex genotype-phenotype maps, where genotypes play the role of parameters and phenotypes the role of trajectories.

3.5.2 Simple Harmonic Oscillators

Although we define macrostates by identifying symmetries underlying general relations, time relations are still of particular interest as a use case because of their long history in physics and their relationship to energy. Here, we demonstrate how MacroNet can automatically identify the symmetry of time translation invariance associated to energy, using a simple harmonic oscillator (SHO) as a case study. The Hamiltonian of the SHO is:

$$\mathcal{H} = \frac{p^2}{2m} + \frac{1}{2}kx^2 \tag{3.8}$$

In this experiment, I let $m = 1$ and $k = 1$ for all cases. The micro-to-micro relation is a temporal relation, represented by pairs of (x_0, p_0) and (x_τ, p_τ), where x_0 and p_0 are the initial position and momentum and τ is uniformly sampled time interval $(0, 2\pi)$ (see Figure 9A). Since I am trying to find a time invariant quantity, I force the two neural networks φ_u and φ_v to share the same weights.

Figure 9 shows our training results. When require the neural network to learn a 1D invariant as a macrostate, the macrostate is exactly a function of energy (Figure 9B). Figure 9C shows samplings from macrostates to microstates. The same color represents microstates sampled from the same macrostate. The sampling shows how the neural

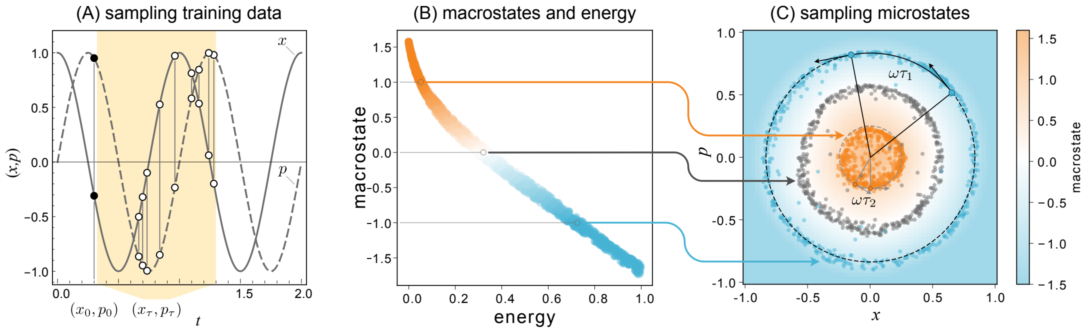


Figure 9. With a Simple Harmonic Oscillator, I Train a Neural Network to Find Invariant Quantities as a Special Case of Macrostates. (A) The (u, v) Pairs Are Sampled from Simulations, Where $u = (x_0, p_0)$ (the Black Dots) and $v = (x_\tau, p_\tau)$. The τ Is Sampled from a Uniform Distribution $U(0, 2\pi)$. The White Dots in the Yellow Region Show a Sampling Example of v. Due to the Randomness of τ, It Is Impossible for Accurate Prediction at the Micro Level. (B) The Neural Network Learns a Function of Energy as the Invariant Quantity. The x-Axis Is the Energy of Microstates Computed by the Physical Theory of SHOs Discovered by Humans, and the y-Axis Is the Macrostate Discovered by the Neural Network. They Show a Monotonical Relation, Which Implies the Successful Identification of Energy by the Neural Network. (C) Conditional Sampling Microstates from $P((x, p)|\varphi(x, p) = \alpha_i)$, Where the α_i Are the Given Macrostates. The Results Approximate Equal Energy Surfaces, Denoted by the Dashed Circles. Note That the Noise in the Sampling Is a Side Effect of the Noisy Kernel Trick I Use Here. The Background Color Also Shows the Learned Macrostate Mapping as a Field.

network has identified three concentric circles, which correspond to the equal energy surfaces of the SHOs (Figure 9C), where the equation $p^2 + x^2 = \mathcal{H}$ represents a circle with a radius of $\sqrt{\mathcal{H}}$. Note that the uncertainty of τ makes it impossible to accurately predict the future microstates. In fact, the optimal prediction at any microstate will be zero when optimizing the MSE loss. However, using MacroNet, it can still predict the future macrostates and sample microstates from them. This is an example of how predictions at the micro level can fail, and how macrostates can help solving many-to-many mapping problems, such that predictions are still possible.

3.5.3 Turing Patterns

Finally, I applied the same method on a complex system: Turing patterns. Here, I use the Gray-Scott Model (Gray and Scott 1984), a 2D space that has two kinds of components, a and b, which might, for example, correspond to two different kinds of chemical species. The a and b are two scalar fields corresponding to concentration of the two species. Their dynamics can be described by the differential equations:

$$\frac{\partial a}{\partial t} = D_a \nabla^2 a - ab^2 + F(1-a) \tag{3.9}$$

$$\frac{\partial b}{\partial t} = D_b \nabla^2 b + ab^2 - (F+k)b \tag{3.10}$$

where D_a, D_b, F and k are four positive constants - these four parameters determine the behavior of the system. This model can generate a set of complex patterns, see Figure 10A. By finding macrostates shared by patterns and parameters, we can then in turn design related systems by sampling parameters that will yield user-specified patterns. Here, u is the parameter vector, $u = (D_a, D_b, F, k)$. And v is the generated pattern, represented by 64×64 images, $v = (a^{(64\times64)}, b^{(64\times64)})$.

We trained the neural network to map parameters and patterns to each other at the macro level (such that these will share the same macrostate). Figure 10 shows the sampling based on the specified patterns. By giving an example pattern v_e (Figure 10A), we can sample parameters $u' \sim P(u|\varphi_u(u) = \varphi_v(v_e))$ with the same macrostate as v_e (Figure 10B). As Figure 10C shows, the sampled rules (set of four parameters) will generate patterns similar to the example patterns. This experiment shows that our method can design complex systems by sampling parameters that will

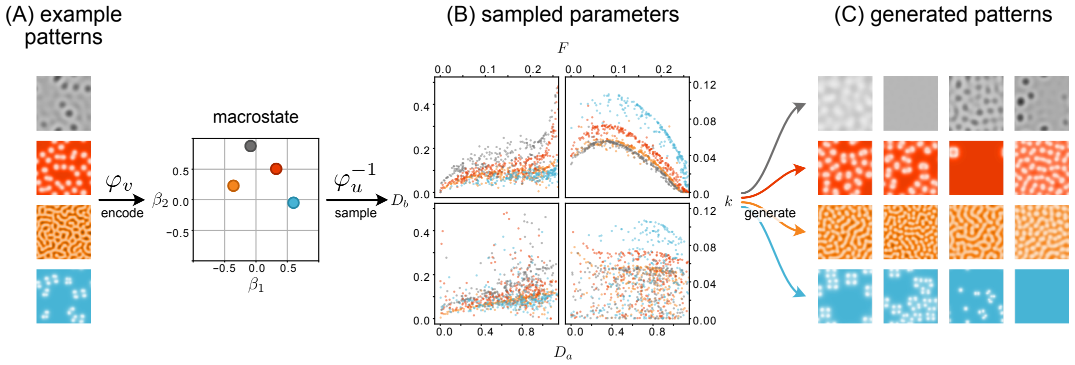


Figure 10. Experiments on Turing Patterns. (A) By Giving an Example Pattern, We Can Compute Its Macrostate by φ_v. The Patterns Are Colorized for Distinguishing Different Experiments. (B) Then, We Can Sample an Ensemble of Corresponding Parameters from the Macrostate by φ_u^{-1}. The Points with the Sample Color Are Sampled from the Same Macrostate Computed from the Corresponding Example. (C) Using the Sampled Parameters, We Can Generate Turing Patterns with Sampled Initial States. The Generated Patterns Show Similar Macroscopic Shape as the Example Patterns.

generate patterns exhibiting the same macrostate as the example behavior. That is, MacroNet can solve the inverse problem of going from pattern to parameters.

The microstate ensembles associated with macrostates can also be directly discovered by this approach. Figure 10B shows the distribution of parameters sampled from different macrostates. The sample points with the same color are considered as equivalent to each other under the mapping φ_u, which takes the microstate to a macrostate. Parameters in the same equivalence class (sharing the same symmetry) will therefore lead to patterns that have the same macrostates, so we can sample any parameters along these equivalence curves and generate Turing patterns with the user-specified behavior.

An additional feature is that observing the sampled parameters can also tell us the importance of different parameters for specifying a target macro behavior. For example, as shown in Figure 10B, different macrostates have similar sampling on D_a.

However, on (F, k), different macrostates sample different parameters. This indicates that F, k will have stronger effect on differences in macro behavior than D_a. This has implications for specifying control parameters in controlling and designing complex systems. An example of interest is in pattern formation in regeneration (Levin 2014), where a framework like MacroNet could identify the patterns controlling specific features of shape.

3.6 Discussion

Since Anderson published the seminal paper, More is Different, it has been increasingly recognized that complex systems displaying emergent behaviors do not necessarily share the same symmetries as their micro-rules (Anderson 1972; Strogatz et al. 2022). That is, we know the mapping from a underlying rule to a large-scale system does not preserve all the symmetries of the underlying rule, due to symmetry breaking and perturbations from the environment. In some sense, this is the very definition of "emergence". However, we might expect some symmetries to be retained such that micro-rules share at least a subset of their symmetries with any macroscopic emergent behavior. Indeed, this is what we see in the experiments presented in this work. Each macrovariable can represent a type of symmetry: for instance, the energy of a simple harmonic oscillator represents how all states with the same energy are symmetric in time to others with that energy. In a more complex case, the macrostates of Turing patterns contain the information that is invariant under the mapping from parameter to pattern, even under external perturbations. The parameters that have the same macrostate are symmetric to each other because they all generate the patterns with the same macrostate. By finding the macrostates via the mutual information shared between ensembles of microstates, we can find the symmetries shared by the two sets of microvariables. This is a general framework for identifying macrostates as maps conserving the symmetries of systems: hence, while given "more is different" is true in most cases, we can still find examples of macrovariables that behave as "more is same" because they will retain underlying symmetries present at the microscale.

The process of finding macrostates can be considered as a prediction problem: that is, it is one of finding predictable variables of two related observations. There are no such variables if two observations have zero mutual information. Thus, if two observations have non-zero mutual information, we can use macrovariables (ensembles of microstates) to connect the two observations. In this way, one can consider macrostates as the instantiated mutual information mapping observations of one system to another (or a system to itself at a different point in time).

Across our experiments, we showed how macrostates can emerge from identifying predictive relations between two sets of observations. The parameter-trajectory relation leads to the macrostate of rotation and direction. The temporal relation between past and future leads to the macrostate energy in the simple harmonic oscillator. In the more complex case of Turing patterns, macrostates arise from parameter-pattern relationships. Thus, by adopting this relationalism idea, we can establish an approach targeting an ambitious question in the complex systems field: is it possible find general laws of complex systems? To address this question, one key task is to find a set of universal macrostates that can be found in most complex systems. If such universal macro level behavior could be identified, the laws of the universal macrostates would be considered as the general laws of complex systems. The method proposed in this work makes an initial step for this ambitious target – by finding macrostates from relations, the macrostates can be used on both sides of the relations (although they may be interpreted differently on either side of the relation). For instance, in the Turing pattern case, the macrostates are not only the macrostates of patterns, but also the macrostates of parameters, and indeed these are both one in the same because of the conserved symmetry. For future work, to find more universal macrostates, the framework may be extended from a second-order relationship to higher-order relationships. Applying this method more generally to complex systems may reveal there are indeed universal general laws, or it may reveal that no map can apply to all systems – that is, that the laws of complex systems are unique to specific classes of systems. In either case, the framework we have presented here, which offers an automated means for identifying general laws via symmetries in complex systems, offers new opportunities for asking and answering such questions.

3.7 Acknowledgement

This project was supported by a grant from the John Templeton Foundation. I would like to acknowledge Dr. Cole Mathis, Dr. Daniel Czegel, Dr. Douglas G. Moore and Dr. Enrico Borriello from the Walker lab at Arizona State University for suggestions and discussions throughout various stages of the work. I thank Prof. Yi-Zhuang You from UCSD for his valuable suggestions on machine learning, Prof. Yunbo Lu from Tongji University for his support on computational resources for part of the project, and Prof. Leroy Cronin from University of Glasgow for helpful discussions and research support.

Chapter 4

UNCOVERING GENOTYPE-PHENOTYPE MAPPING IN COMPLEX CHEMICAL SYSTEM BY IDENTIFYING MACROSTATES

4.1 Abstract

In biological systems, specific genes are associated with certain features. This mapping from gene to feature is often referred to as genotype-phenotype mapping. Notably, this concept extends beyond the realm of biology, with analogous relationships observed in domains such as rule-pattern mapping and chemical compound-behavior mapping. However, understanding genotype-phenotype mappings poses significant challenges. The assumption of a direct one-to-one correspondence between genotypes and phenotypes is often overly simplistic. In reality, multiple genes can collectively influence a single phenotype, while a single gene can also affect multiple phenotypes. Additionally, given the reality that many phenotypes cannot be controlled and numerous genotypes do not regulate any characteristic, the quest for regularities within these higher-order relations frequently leads to an exponential surge in potential combinations. This factor considerably exacerbates the complexity of the task. Nevertheless, by identifying concealed macrostates within both genotypes and phenotypes, these intricate high-order regularities can be simplified through macro-variables. In this study, we have elected to focus on the oil droplet movement system as an example of a chemical genotype-phenotype mapping system. This system involves oil droplets of varying chemical compositions exhibiting different movement patterns within a Petri dish. By observing these patterns and identifying macrostates, we discovered

two independent dimensions. Each dimension reveals how a combination of chemical compounds governs a specific aspect of the oil droplet movement patterns. Employing a novel regression method proposed in this study, we were able to extract concise interpretable regularities from observations of this complex chemical system. Furthermore, an experimental chemical validation was conducted to validate the model prediction. One of the macrostates was successfully controlled.

4.2 Introduction

In biological systems, DNA or RNA predominantly controls the characteristics of an organism. Different genotypes often result in different phenotypes, similar to how different book titles typically correspond to different contents. This type of relationship is widely observed. For example, consider the rule-pattern mapping in elementary cellular automata: certain rules may produce blank patterns, while others lead to more complex ones (Wolfram et al. 2002). Intriguingly, some rules with divergent initial states can elicit completely dissimilar behaviors (Riedel and Zenil 2015). Another instance is the parameter-behavior mapping in Turing patterns (Gray and Scott 1984). In reaction-diffusion systems, a set of parameters can lead to similar or differing behaviors (Pearson 1993). Broadly speaking, these mappings are all analogous to genotype-phenotype mapping, exhibiting comparable characteristics and posing similar challenges.

Mathematically, genotype-phenotype mapping can be conceptualized as associations between strings or vectors (representing genotypes) and structures (representing phenotypes). The definition of phenotype is typically quite broad, including patterns at all levels of resolution as well as behaviors (Ahnert 2017). A critical aspect of such

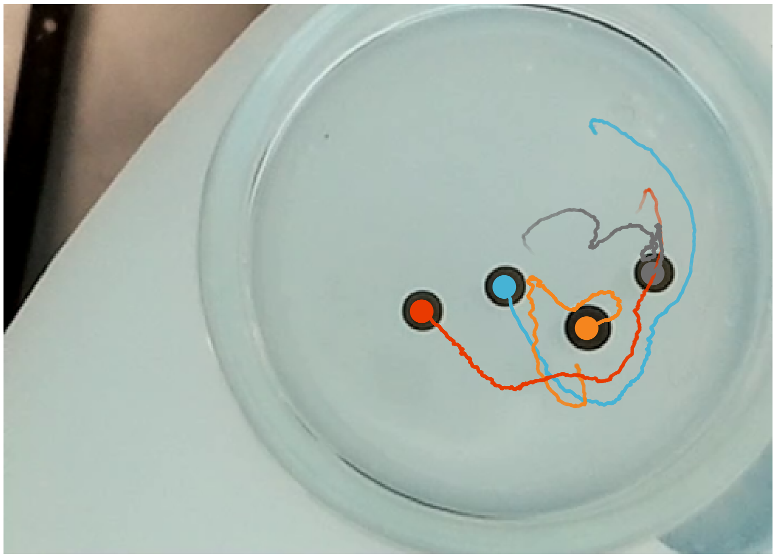


Figure 11. The Movement Trajectories of Oil Droplets in a Petri Dish Are Indicated by Lines of Various Colors. Distinct Configurations of Chemical Compounds Will Result in an Array of Unique Movement Patterns.

mapping is its many-to-many nature. In other words, the same genotype, subject to different boundary conditions and external noise, can often lead to multiple different phenotypes. Conversely, differing genotypes can also yield the same phenotype due to neutral mutations or external noise. This factor poses a challenge to answering a fundamental question: which genotype(s) control which phenotype(s)?

Considering the complexity of biological systems and their lengthy experimental timespans, alongside the oversimplified models that can often be too ideal, we focus our study on the intriguing behavior exhibited by oil droplets in Petri dishes (see Figure 11) when combined with four distinct chemical compounds (Gutierrez et al. 2014). These droplets display compelling movement patterns, ranging from static positioning and motion to more complex activities like splitting or merging. And these movement patterns can be controlled by chemical compounds via evolutionary selections. Hence, this serves as an advantageous platform to study genotype-phenotype mapping, with

$$\begin{tikzcd}
\alpha_i \arrow[r, dash, "{P(\alpha,\beta)}", "\text{mutually predictive}"'] & \beta \\
\text{parameter} \arrow[u, "\varphi_u"] \arrow[r, "\text{data pairs}"] & \text{trajectory} \arrow[u, "\varphi_v"]
\end{tikzcd}$$

Figure 12. Illustration of Identifying Macrostates and Design Parameters. Both Parameter and Trajectory Are Been Mapped to Macrostates That Are Been Trained to Be Mutually Predictive.

our primary focus being the relationships between the chemical compounds and the resultant behaviors.

One of the major challenges within this scope is determining which aspects of the chemical compounds can predict specific behavioral elements, as previously mentioned. For instance, consider a compound that influences the oscillation magnitude of the oil droplets. Traditional predictive models might struggle in this scenario as they generally aim to predict average trajectories. However, averaging trajectories with different oscillation phases would produce a smooth trajectory devoid of oscillation. Difficulties can also arise when moving in the inverse direction, from the desired movement patterns to chemical compounds, where different compounds can result in similar movement patterns. These issues fundamentally arise from the many-to-many mapping nature inherent in genotype-phenotype relationships.

These complexities render both parameter design and trajectory control difficult due to the uncertainty of predictive relationships. To mitigate this, we introduce a framework for identifying macrostates as mutually predictive information (Zhang and Walker 2022). This methodology allows us to extract macrostates as mutually predictive information from both chemical compounds and trajectories (see Figure 12). In turn, when configuring parameters for specific trajectory types, we can sample compounds from a given macrostate.

From an information perspective, we abandon attempts to predict unpredictable details. Rather, we focus on predicting the macrostate (the predictive information) and sample unpredictable details. This approach avoids the averaging issue discussed previously.

It is essential to select an appropriate dimensionality of macrostates such that the macrostates have sufficient mutual information between genotypes and phenotypes. Enough mutual information is crucial for explicating each dimension of the macrostates. To do this, based on the MacroNet architecture (Zhang and Walker 2022), we propose a method for estimating mutual information between the chemical compounds (parameters or genotypes) and oil droplet trajectories (patterns or phenotypes). This method helps us determine the quantity of mutual information that can be conveyed by a certain number of macro-variables, enabling us to select an appropriate dimension of the macrostate for the oil droplets.

4.3 Method

4.3.1 Oil Droplet and Data Pre-processing

Our experiment involves the use of oil droplets created from four chemical compounds: octanoic, DEP, pentanol, and octanol. For each experiment, we placed four droplets into a Petri dish and captured videos of their behaviors. To accumulate ample training data, we employed an autonomous system capable of repetitively conducting this experiment, each time testing different ratios of the chemical compounds. 900 unique experiments were conducted, each involving six parameters: the ratio of the four chemical compounds along with humidity and temperature data.

We preprocessed the video data by tracking the trajectories of the oil droplets and recording their size and shape details. Given that the oil droplets can split, merge, or even dissolve in the Petri dish, despite beginning each experiment with four droplets, we refrained from pairing the genotype-phenotype by chemical compounds and behavior of all droplets—owing to the fact that the total droplet number can escalate to approximately 50. Instead, we constructed the genotype-phenotype pairs by associating chemical compounds with single droplet trajectories. To maintain simplicity, we prioritized droplets demonstrating prolonged stability, thereby filtering out those of notably small size.

Initially, the raw video data were captured at a rate of 150 frames per second. To manage data volume while minimizing significant information loss, we resampled the oil droplet trajectories to 10 frames per second. Subsequently, around 1700 droplet trajectories were selected for training. Our primary filtering process focused on several attributes such as the trajectories' initial frame, length, number of outliers, and shape. Secondary filtering solely considered trajectory length to accommodate different training strategies more effectively.

Given the smooth trajectories exhibited by these oil droplets, we employed an auto-encoder for dimensionality reduction as a pre-training process, thus accelerating the subsequent training phase. We trained InfoVAE (Zhao, Song, and Ermon 2017) models to encode the trajectories (time series of positions and sizes) into a 32-dimensional format. Following this, these 32-dimensional trajectory representations, along with the 6-dimensional parameters, were mapped into macrostates of lower dimensions using another neural network.

4.3.2 Identify Macrostate by Normalization Flow

Given our need for both encoding and generating, we employ MacroNet to identify macrostates from parameter-trajectory pairs. MacroNet utilizes two encoders based on normalization flow (or flow-based models), which map parameters and the compressed trajectories into a lower-dimensional vector. Subsequently, we train these two vectors to be identical to each other. This training objective allows the encoders to identify the mutually predictive information that can be mirrored in both chemical parameters and trajectories. Since we're using normalization flow to construct our encoder, we can reverse the encoding process and sample the chemical compound configurations from a given encoding or, as we prefer to term it in this paper, macrostate. This encode-generate architecture refrains from directly predicting parameters or trajectories. Instead, it predicts only the predictable information, which is the macrostate encompassing information on microstate distribution.

Contrastive training objectives often lead to the mode collapse problem, where the encoders simply output a constant value or a lower-dimensional manifold. This issue is commonly addressed by incorporating a large number of negative samples. However, the normalization flow models offer a streamlined solution. With normalization flows, we can directly estimate the probability density by computing the log-determinant of the Jacobian for the encoder. Thus, we can directly optimize the outputs to follow a certain distribution. In this case, we train the output to follow an independent normal distribution. Enforcing this particular distribution helps to prevent trivial solutions, such as constant numbers or lower-dimensional manifolds, as outlined in Chapter 3. Moreover, the normal distribution simplifies the estimation of mutual information, thereby laying the groundwork for the subsequent mutual information estimation methodology. As a result, the training objective becomes:

$$\theta = \arg\min_{\theta} \mathbb{E}_{u,v\sim P(u,v)} \left(\|\varphi_u(u) - \varphi_v(v)\|^2 - \mathcal{L}_D(u) - \mathcal{L}_D(v)\right) \tag{4.1}$$

where φ_u and φ_v are encoders for chemical parameters and trajectories, respectively. The $\mathcal{L}_D$, or distribution loss, can be computed at a low cost when using normalization flows:

$$\mathcal{L}_D(u) = -\frac{\|\alpha'\|^2}{2} + \log\det\frac{\partial\alpha'}{\partial u}. \tag{4.2}$$

Here, α' is the original output of the normalization flow. Due to the invertibility requirement of the normalization flow, α' will have the same dimension as the input vector. To reduce the dimension, some dimensions must be omitted. Hence, the encoder output is $\varphi_u(u) = \alpha = \alpha'_{1:n}$, where n is the macrostate dimension. When performing inverse sampling, the first n elements are given, while others are sampled from an independent normal distribution. This approach allows us to achieve our encode-generate objective.

4.3.3 Mutual Information Estimation

In the process of identifying mutually predictive variables as macro-variables, we are inherently required to answer two principal questions: What is the efficacy of the training? And subsequently, how many macro-variables should be selected?

Though we utilize two loss functions as training objectives, namely prediction loss and distribution loss, they do not provide an impeccable gauge of effectiveness. The prediction loss merely demonstrates the proximity between the predicted macrostate and the actual macrostate without indicating the distribution. Similarly, the absolute value of the distribution loss fails to yield substantial insights about the distribution. Thus, the outcomes of these training objectives offer no clarity on the quality of the results. Consequently, it necessitates a direct estimation of the mutual information between the macrostates on either side. This estimation will enable us to tell how much information is shared between chemical parameters and the trajectories.

Estimating the mutual information between two random variables, particularly in the context of high-dimensional variables, is typically a complex task that has been the focus of numerous research efforts (Kraskov, Stögbauer, and Grassberger 2004; Belghazi et al. 2018). However, in our specific case, the task is considerably simplified because we train the macro-variables to adhere to independent normal distribution. As a result, the estimation of mutual information becomes relatively more straightforward.

If we view the macro-variables on either side of the paired data as two random variables, α, and β, their mutual information can be computed as the difference between the summation of their entropy and the entropy of their joint distribution, which is represented as follows (Cover and Thomas 1991):

$$I(\alpha, \beta) = H(\alpha) + H(\beta) - H(\alpha, \beta) \tag{4.3}$$

Since the trained macro-variables will follow normal distributions with covariance matrices, $N^n(\mu, \Sigma)$, their differential entropy is relatively simple to compute (Lazo and Rathie 1978):

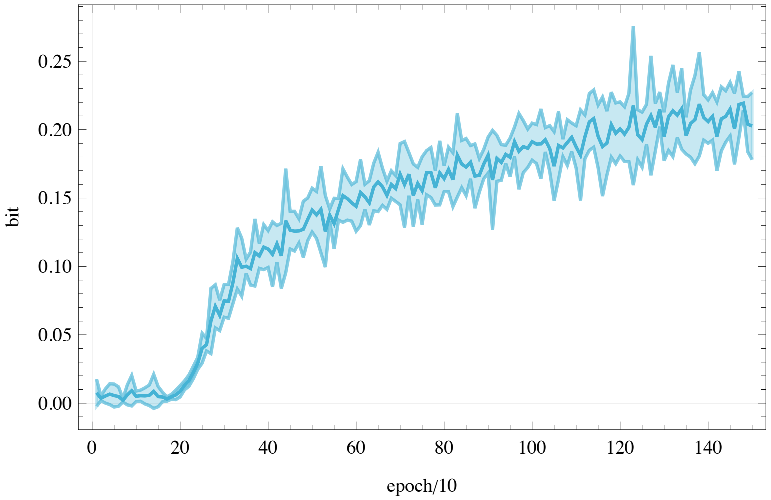


Figure 13. The Estimated Mutual Information of the Macro-variables Increases Through the Training Process. The Background Shadow Indicates the Standard Deviation of Mutual Information Estimated from Different Trainings of Cross-validation. The Covariance Matrices Are Computed Using 256 Samples at Each Point, and the Cross-validation Employs a k-fold Method, with k Set at 5.

$$H[N^n(\mu,\Sigma)] = \frac{1}{2}n + \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln\det\Sigma, \tag{4.4}$$

where n is the dimension of the random variable, and Σ is the covariance matrix. Hence, the mutual information can be computed as:

$$\begin{aligned} I(\alpha,\beta) &= H[N^n(\mu_\alpha,\Sigma_\alpha)] + H[N^n(\mu_\beta,\Sigma_\beta)] + H[N^n(\mu_{\alpha+\beta},\Sigma_{\alpha+\beta})] && (4.5)\\ &= 2\left(\frac{n}{2} + \frac{n}{2}\ln 2\pi\right) + \frac{1}{2}\ln\det\Sigma_\alpha + \frac{1}{2}\ln\det\Sigma_\beta - \frac{1}{2}\ln\det\Sigma_{\alpha+\beta} - (n + n\ln 2\pi) && (4.6)\\ &= \frac{1}{2}\ln\left(\frac{\det\Sigma_\alpha\det\Sigma_\beta}{\det\Sigma_{\alpha+\beta}}\right) && (4.7)\end{aligned}$$

The covariance matrices can be empirically computed with ease. By leveraging this metric, we can quantify the quality of the training result. Our experiments reveal that mutual information bears a significant correlation with the quality of the macrostate. Training results for macrostates with higher mutual information manifest a superior correlation with human data comprehension. Throughout the training process, an increase in mutual information is observed, suggesting that MacroNet effectively extracts the mutually predictive variables from paired data (see example in Figure 13). It's worth noting that this method hinges on the assumption of normal distribution. Early values may not be entirely reliable because the outputs need time to approximate a normal distribution. Consequently, in practical applications, a minor degree of manual selection for an early stopping point (Prechelt 2002) is often required when choosing a model for subsequent experiments.

4.4 Results

4.4.1 Mutual Information and Dimensions of Macrostates

A crucial question in our study pertains to the optimal dimensionality of macrostates. Ideally, this should not exceed the minimal dimension of the parameters and trajectories. In this context, the maximum dimension would be six, corresponding to the parameter size. However, not all parameters can independently control the trajectory behaviors, and not all trajectory details can be influenced by parameters. Therefore, we anticipate the existence of an optimal dimension for macrostates wherein most of the mutual information between parameters and trajectories is encapsulated.

The method of mutual information estimation we introduced can address this dimensionality question. For a given dimension of a macrostate, the model will achieve a specific value of mutual information. Through optimization of the neural network

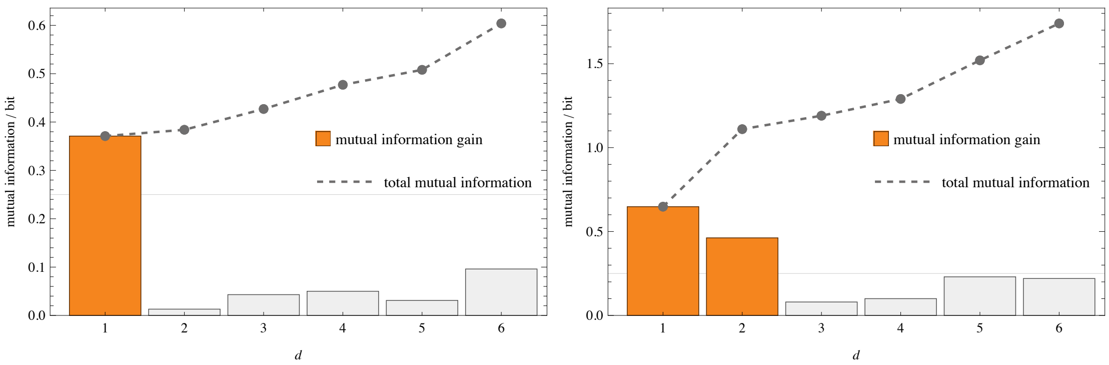


Figure 14. The Maximal Mutual Information Conveyed by Macrostates at Different Dimensions. Left: Training the Macronet with Weight-fixed InfoVAE. Right: By Continually Training the InfoVAE Encoder, We Can Capture More Mutual Information with Macrostates.

for peak performance, this mutual information can approximate the maximum mutual information that can be conveyed by this dimensionality.

As the number of macrostate dimensions increases, so does the mutual information. The extent of this increase corresponds to the additional mutual information that the added dimension can carry. By setting a threshold, we can objectively ascertain the number of macrostates present in a system. Given the inherent randomness in both training and mutual information estimation, we employ k-fold ($k = 5$) cross-validation five times for each dimension, extending across 3000 epochs. Notably, when the dimension is set to one, we extend the training to 10,000 epochs as this configuration demonstrates a slower pace in achieving peak performance.

We carried out experiments on mutual information at different dimensions using two training strategies. In the first strategy, we initially trained an InfoVAE (Zhao, et al, 2017) to disentangle various features of trajectories into 32 dimensions. Subsequently, we froze the encoder and incorporated a linear invertible layer (Kingma, et al, 2018) to further decrease it to the dimension of the macrostate. As depicted in Figure 14(left), increasing dimensionality does encapsulate more mutual information since the first dimension captures a significant portion, specifically 61.4% of mutual information.

Our second strategy diverges from the first in that we do not freeze the encoder; rather, we concurrently train it with the linear invertible layer. This method allows the identified macrostates to encapsulate more mutual information, which continues to rise as we increase the macrostate dimension (as depicted in Figure 14 (right)). Under this framework, the most mutual information is conveyed by the first two dimensions, while additional dimensions contribute less. We implemented a threshold of 0.25 bits to ascertain the optimal dimension. Although increasing the dimension augments the total mutual information conveyed by the macrostate, the additional dimension also injects more randomness into the macrostate space due to its low mutual information contribution, thus increasing entropy. To maintain macrostate interpretability with minimal entropy, while preserving most mutual information, a threshold is essential. Based on this framework and threshold, we set the macrostate dimension at two.

Upon examining the trajectory embedded in the macrostate space, we observed that while the first strategy yields a clearer intuitive trend, it encapsulates less mutual information. The second strategy, though rendering it more challenging to discern the significance of different macrostate dimensions, does not adversely affect our goal of designing chemical parameters. As depicted in Figure 15, the first strategy, despite providing a more intuitive trajectory embedding, captures less mutual information with macrostates. This could potentially make parameter design less informative. Therefore, we have elected to proceed with the second strategy for further study. Importantly, as referenced in the mutual information section, mutual information estimation may exhibit a bias in the early stages. To mitigate this, we manually select

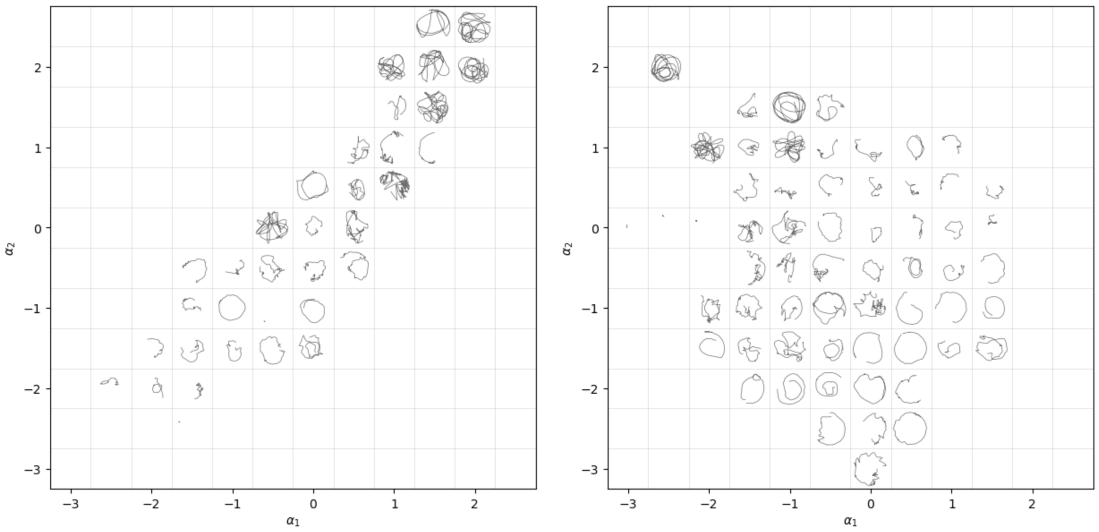


Figure 15. The First Strategy (Left) Provides a Clear Meaning at Different Macrostate Dimensions. Conversely, the Second Strategy (Right), Despite Presents a Less Intuitive Trend Across Each Dimension, Identifying Macrostates with Higher Mutual Information.

a checkpoint from our five training runs at various epochs, which helps to secure a better trend and feature disentanglement.

The distinction between these two strategies underscores the fundamental difference between our approach and other feature disentanglement methods such as VAE or InfoVAE. While certain features, such as high-frequency patterns or minor details, might be overlooked by models primarily aimed at reconstructing micro-details, these very features could serve as important dimensions of macrostates that carry mutual information in our method. This contrast indicates the unique advantage of our approach in the realm of feature extraction.

4.4.2 Understanding Dimensions of Macrostates

As we have two coarse-graining functions for both the chemical and trajectory data, and despite their close training proximity, these functions possess distinct forms. Consequently, it's necessary to understand the dimensions of macrostates for both aspects separately.

In regard to the chemical compound parameter side, given their low dimensions, symbolic regression can be employed to gain a concise understanding. On the trajectory side, due to the high dimensionality, we may need to employ a hypothetico-deductive method to decipher the meaning.

4.4.2.1 Macrostate and Chemical Compounds

Heuristically, by doing experiments, a linear fit can reveal substantial information about how parameters are mapped to macrostates. However, as the MacroNet does not have any sparse coding requirements, a direct regression might not show a correlation between chemical compounds and macrostates with a sufficiently sparse result. More specifically, due to the rotational symmetry of the macrostate embedding, a rotation might exist that allows macrostates to have a sparser relation to the chemical parameters. To accomplish this, we propose a simple modification to linear regression: instead of fitting a linear function f such that $y = f(x)$, we also fit a rotation matrix R_θ, to achieve $R_\theta(y) = f(x)$. We apply L1 regularization (alpha=0.1) only to the coefficients of the function f. To implement this concept, we use EUNN, a parameterized unitary matrix, to represent R_θ (Jing et al., 2017), and a linear layer to represent f, subsequently training it via the SGD method. By using this method,

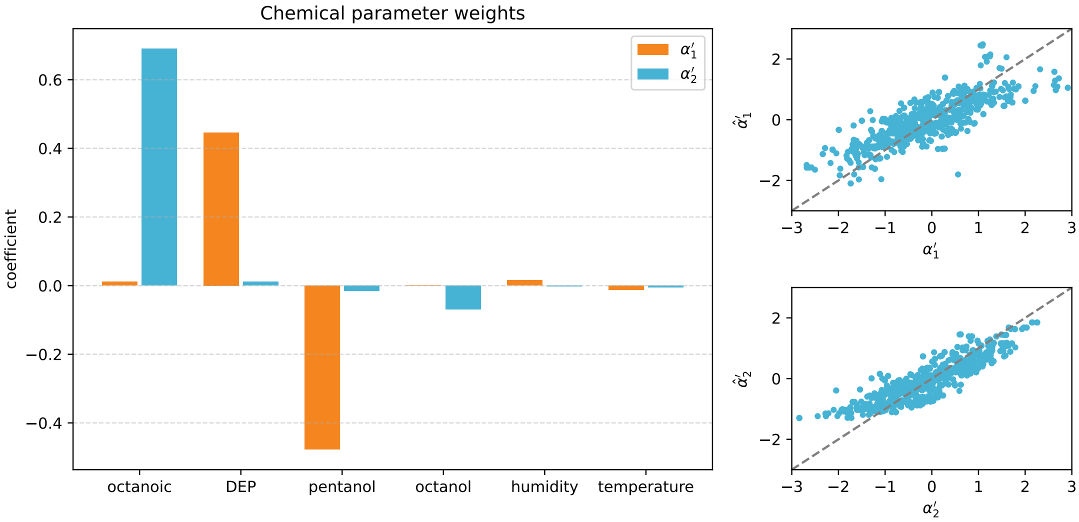


Figure 16. The Linear Regression Result Fitted Alongside a Rotation Matrix. The Coefficient Indicates the Degree to Which the Concentration of Different Chemical Compounds Contributes to the Two Dimensions of the Rotated Macrostates. Higher Absolute Values of the Coefficient Imply Greater Contributions. The Two Sub-figures on the Right Demonstrate a High Correlation Between the Fitted Macrostates, Denoted as $\hat{\alpha}'$, and the Original Macrostate, α'.

we discerned two almost orthogonal dimensions from the chemical space (refer to Figure 16): the first dimension is determined by the ratio difference between DEP and pentanol, while the second dimension is determined solely by the ratio of octanoic.

Consequently, the rotation matrix is:

$$R_{\theta=0.93} = \begin{bmatrix} 0.598 & 0.802 \\ -0.802 & 0.598 \end{bmatrix} \tag{4.8}$$

The trajectory embedding post-rotation is displayed in Figure 17. In this rotated embedding, each dimension attains enhanced interpretability. According to this figure, a higher DEP and lower pentanol lead to a more “complex” movement pattern, which traverses the entire Petri dish. Additionally, higher octanoic appears to make the trajectory more rugged and localized. Nonetheless, this interpretation is derived

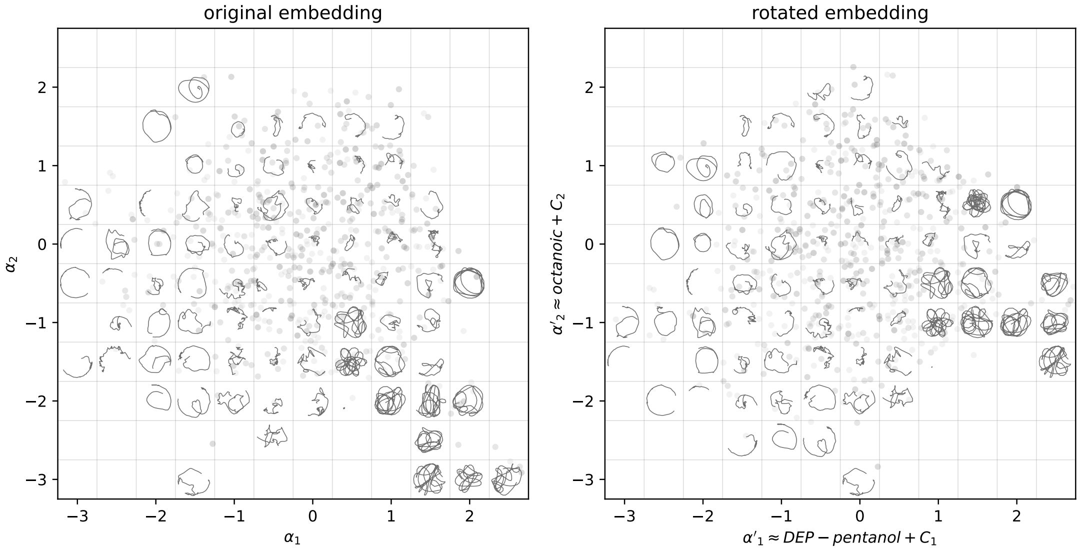


Figure 17. Left: The Trajectory Macrostate Embedding Without Rotation; Right: The Trajectory Macrostate Embedding after Apply Rotation.

from human observation with intuitive explanations. Despite the linear model fitting the macrostate quite effectively, it doesn't imply that we can use the linear model as the first step and bypass the MacroNet training. This stance is supported by two main reasons: Firstly, the model should possess adequate power to carry the mutual information if the data pairs share high mutual information, which is critical in determining the dimension of the macrostate. Secondly, in order to estimate the mutual information accurately, we need the macro-variables to follow a multi-normal distribution. The transformation into such a distribution demands a substantial representational capacity that linear models are unable to provide. As we need to compare the mutual information across different macrostate dimensions, achieving optimal performance is crucial.

4.4.2.2 Macrostate and Trajectories

Name	**Description**	**Definition**
`entropy`	Entropy of the droplet's position distribution.	$\sum_{i,j} p_{ij} \log p_{ij}$, where ij is the index of the bins for counting p_{ij}.
`avg_speed`	Average speed.	$\bar{\lvert v\rvert} = \frac{1}{T} \sum \lVert \mathbf{x}_{t+1} - \mathbf{x}_t \rVert$
`log avg speed`	Logarithm of the average speed.	$\log \bar{v}$
`avg_acc`	Average acceleration.	$\bar{\lvert a\rvert} = \frac{1}{T} \sum \lVert \mathbf{v}_{t+1} - \mathbf{v}_t \rVert$
`avg log size`	Average of the logarithm of the droplet size over time.	$\frac{1}{T} \sum_t \log s_t$, where s_t is the droplet size at time t.
`position_std`	Standard deviation of the droplet's positions over time.	$\sqrt{\sum_t \lVert \mathbf{x}_t - \bar{\mathbf{x}} \rVert^2 / T}$
`size_std`	Standard deviation of the droplet's sizes over time.	$\sigma_s = \sqrt{\sum_t \lvert s_t - \bar{s} \rvert^2 / T}$
`log_size_std`	Logarithm of the standard deviation of the droplet's sizes over time.	$\log \sigma_s$
`size_change`	Average change in size over time.	$\delta s = \frac{1}{T} \sum_t \lvert s_{t+1} - s_t \rvert$
`log_size_change`	Logarithm of the average size change over time.	$\log \delta s$
`size_diff`	Difference in size between $t = 0$ and $t = T$.	$\lvert s_1 - s_T \rvert$
`curvature`	Average curvature of the trajectories.	$\frac{1}{T} \sum \min(k_t, 100)$, where $k_t = \lvert x'y'' - y'x'' \rvert / (x'^2 + y'^2)^{3/2}$

Table 2. Trajectory Features Used in the Linear Regression.

Considering the high dimensionality of trajectories, we initially perform heuristic feature engineering to extract 13 features potentially related to the macrostates (see

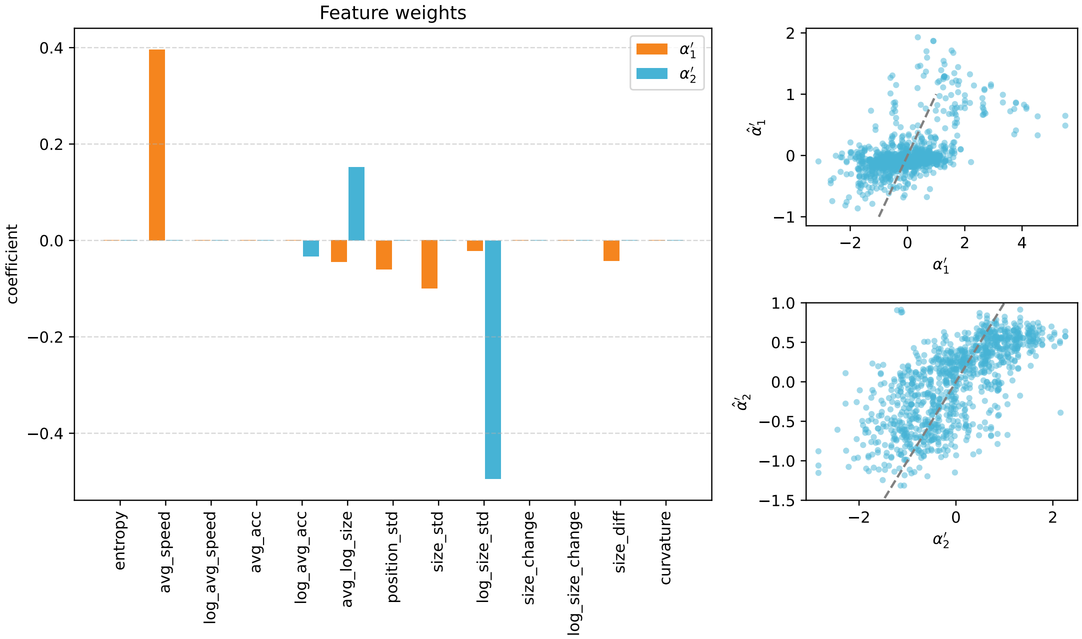


Figure 18. Linear Fitting Coefficient for Trajectory Features and Macro-variables. See Definitions of X-axis in Table 2.

Table 2). By using LASSO (least absolute shrinkage and selection operator) linear regression with a regularization factor of 0.1, we determine that the first dimension of the rotated macrostate correlates with the logarithm of the average speed of the oil droplets. The second dimension, on the other hand, associates with the logarithm of oil droplet size change, specifically the log standard deviation of oil droplet speed (refer to Figure 18).

Upon combining the analysis of both chemical compound concentrations and oil droplet trajectories, we can deduce the presence of two independent macro-variables in this chemical system. The first variable suggests that a higher concentration of DEP in conjunction with a lower concentration of pentanol will enhance the movement speed of oil droplets. The second variable indicates that a higher concentration of octanoic acid will limit the size variation (see `log_size_std` defined in Table 2) when oil droplets are in motion within the Petri dish.

4.4.3 Macrostate Embedding and Parameter Design

Through the identification of macrostates within parameter-trajectory pairs, we can glean insights into the spatial pattern of trajectory macrostates. We present the results of two-dimensional macro-variable embeddings here (refer to the top-right section of Figure 9). By applying the inverse function of the neural network, we can sample chemical parameters capable of generating trajectories corresponding to a specific macrostate. In this case, we sampled 256 parameters for macrostates at coordinates (1,1), (1,-1), (-1,1), and (-1,-1). These particular macrostates were chosen to highlight the effects of varying macro-variables.

The comparison of these distributions confirms our previously stated conclusions. The concentration differential between DEP and pentanol exerts the most significant influence on the first macro-variable, while octanoic concentration correlates with the second macro-variable (refer to the scatter plots and histograms in Figure 19).

4.4.4 Experimental Validations

Following the sampling of microstates corresponding to the four selected macrostates, we executed chemical experiments to validate these macrostates. Rather than sampling from the distribution, we sampled the top five most probable microstates associated with each given macrostate, highlighted in Figure 19. See Appendix B.1

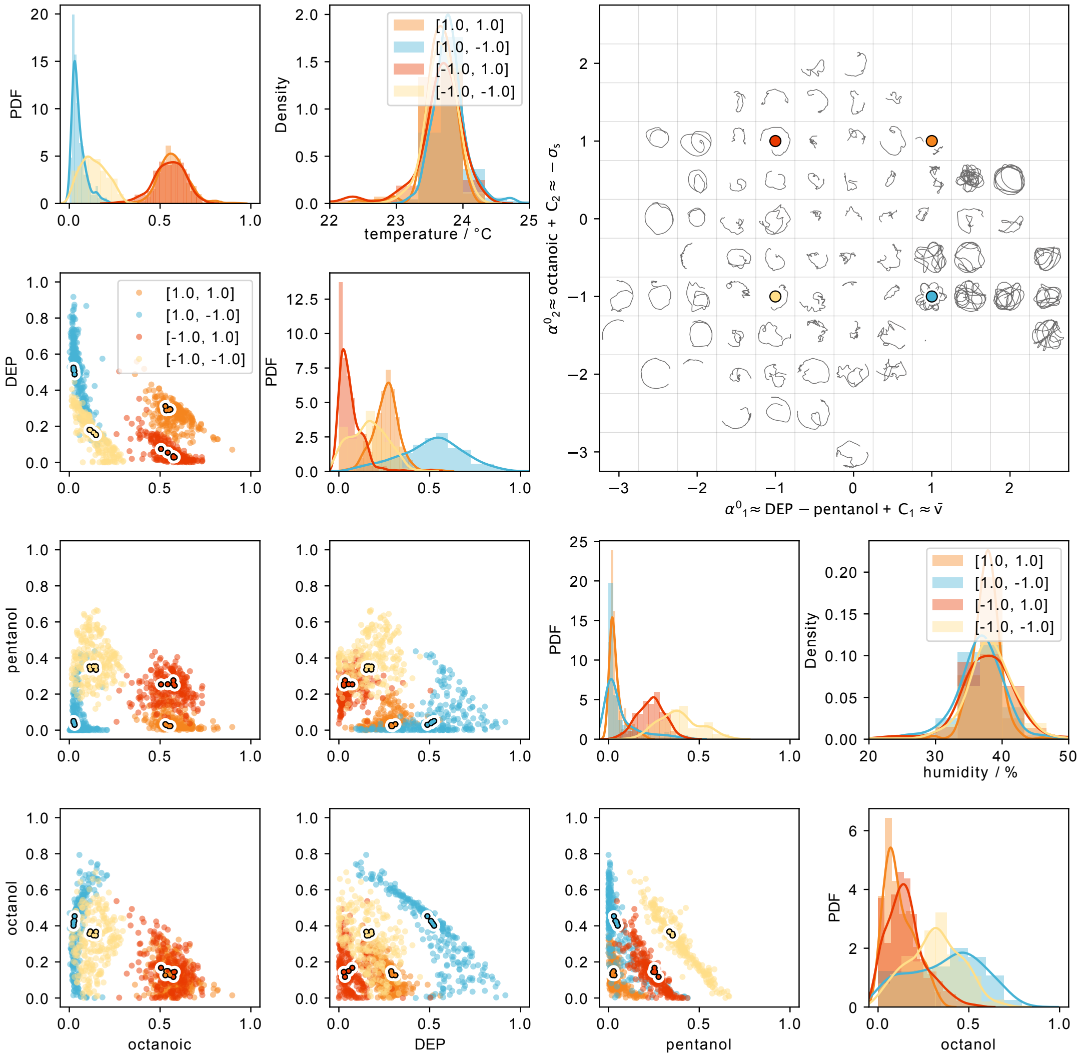


Figure 19. This Figure Depicts the Distribution of Sampled Chemical Parameters Sampled from Different Macrostates. The Scatter Plots in the Triangular Region at the Bottom Left Display Various Chemical Compounds Sampled from These Macrostates, with Each Macrostate Represented by a Different Color. The Highlighted Points Correspond to the Sampled Microstates Used for Experimental Validation. Along the Diagonal Are Figures Illustrating the Marginal Distribution of Each Chemical Compound. In the Top-right Corner, Trajectories Are Embedded in the Rotated Macrostate Space, with Colored Dots Indicating the Macrostates from Which Microstates Are Sampled. The Distribution of Sampled Temperature Is Presented in the Top Middle, and the Sampled Humidity Is Displayed in the Middle Right. Notably, Neither Temperature nor Humidity Shows Significant Variation Across Different Macrostates.

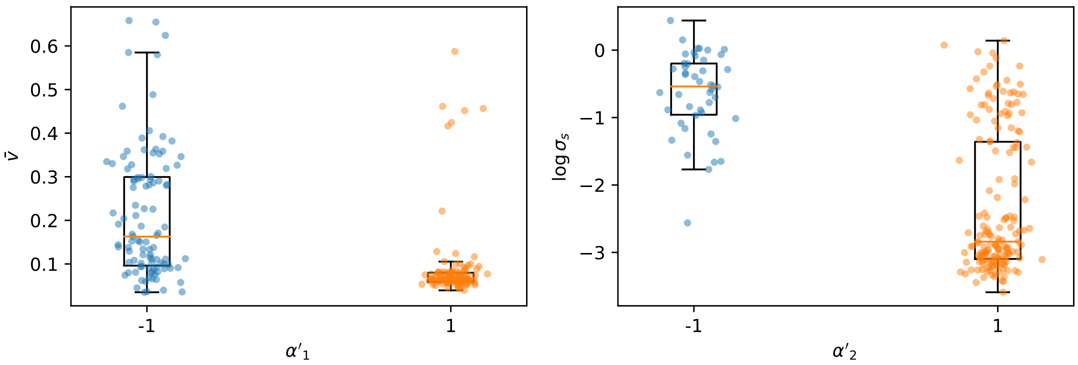


Figure 20. The Figure Displays the Features of Oil Droplet Moving Trajectories under Varying Experimental Parameters, Which Have Been Sampled from Distinct Macrostates (X-axis). The Left-hand Figure Demonstrates a Decrease in Speed with Higher Macro-variable α'_1, Contradicting the Model's Prediction of a Positive Relationship. Conversely, the Right-hand Figure Presents a Reduction in Log Size Change with an Increase in Macro-variable α'_2, Aligning with the Model's Predictions.

for the sampling details. Each sampled microstate was subjected to five replicated experiments. The same video analysis was subsequently applied to extract the moving trajectories of the oil droplets. While we were unable to control temperature and humidity, our preceding study demonstrated that these factors did not significantly impact our results. However, the humidity range was out of the training distribution. Due to seasonal variations, the validation humidity was $49 \pm 1\%$, compared to the $38 \pm 4\%$ humidity of the training data. The temperature also shifted from $23.6 \pm 0.6°\text{C}$ to $24.2 \pm 0.2°\text{C}$. Despite these changes not affecting the macrostate, such deviations that are beyond the training distribution could potentially yield unexpected results. Given their negligible impact on the macrostate, we set all temperature and humidity values to the average values from the training set when computing the macrostate of the validation set.

Figure 20 presents the validation results. We compared the trajectories associated with different microstates sampled from corresponding macrostates. The standard deviation of the droplet size over time was successfully controlled. However, the average speed of the oil droplets was not controlled. The validation results reveal a negative correlation between average speed and the first dimension of the macrostate, while the training data indicate a positive correlation.

4.5 Discussion

Science is multifaceted, encompassing aspects such as data collection and observation. However, a common ultimate goal is the discovery of the laws of nature from observations. An observation is a fusion of these laws and the boundary or initial states. Only with such separation, physical laws can be universal. In other words, the law represents the residual information once the information pertaining to boundary states has been removed (Pattee, Rączaszek-Leonardi, and Pattee 2012).

When we intentionally overlook boundary information, a degree of uncertainty arises. Hence directly predicting microstates in this context might not be the most optimal choice, as the distribution's average may not correspond to the highest probability density. For instance, in a spherical shell distribution, the average lies at the center – a location with zero probability density.

To overcome this challenge, a generative model is required to decouple the law from the boundary condition. Therefore, generative models, although often currently used to generate images or language, could play a pivotal role in scientific discovery. These models predict associated distributions (such as using trajectories to forecast the distribution of chemical parameters) and generate that information (like initial states and boundary conditions) which is otherwise unpredictable.

Our research findings affirm this point: our second training strategy uncovers more macrostates than the first strategy that freezes the pre-trained InfoVAE encoder. Although it does not use the trajectory to predict the chemical parameters, it does aim to find a latent encoder capable of accurately reconstructing the micro details of trajectories. However, as we noted earlier, some details may not be related to laws, but instead associated with boundary conditions. Moreover, some overlooked patterns, though potentially related to the law, may still be small and detailed, leading to their exclusion by the VAE. This oversight could disregard important macrostates and misguide our study. Nonetheless, when we unfreeze the pre-trained model and train it in tandem with MacroNet, we enable the encoder to predict only the distribution, thereby relinquishing the futile attempt to predict microstate details.

In the experimental validation discussed in Section 4.4.4, we identified macrostates with two dimensions; however, only one was successfully controlled. Several factors could contribute to this deviation from prediction. One potential reason could be the differences in experimental conditions. The humidity during validation was significantly higher than that of the training data, which could potentially influence the observed behavior. Another potential reason might be the chosen features, as demonstrated in Table 2. Other features might exhibit stronger correlations with the macrostates and thus may be more effectively controlled. This calls for a more systematic exploration of potential features. The final potential reason could be the limitations of the normalization flow neural networks employed in this study. The requirement for invertibility restricts the representational power of the normalization flow models, which could potentially impact their generalizability. One possible solution might be the concurrent use of contrastive learning and conditional generative models, such as diffusion models (Ho, Jain, and Abbeel 2020). This approach could mitigate the constraints on the representational power of the neural networks and potentially enhance performance.

4.6 Acknowledgement

I express my sincere gratitude to Prof. Lee Cronin and Dr. Abhishek Sharma from the University of Glasgow. Their valuable assistance in conducting the chemical experiments and provision of crucial data laid the foundation for this theoretical research.

Chapter 5

SUMMARY

5.1 Abstract

In this dissertation, the significance of macrostates in uncovering general laws governing complex systems is introduced. I have provided a review of various definitions and perspectives about macrostates, leading to the proposal of a unified perspective on different macrostate theories and methods, and hence proposed the relational macrostate theory. With the support of this theory, I developed MacroNet and demonstrated its application to several complex systems. In this chapter, I will explore further the broader potential of this theory and methodology in various applications. In addition, I will reveal the subtle link between macrostate and computational theories, particularly in relation to the concept of computational irreducibility. In other words, macrostates can also be interpreted as the computationally reducible aspects of a system. Moving beyond this connection, I will engage in a discussion on how to enhance the interpretability of macrostates through the integration of innovative machine learning techniques.

5.2 Summary

Throughout this dissertation, I have introduced the Relational Macrostate Theory and developed a machine learning architecture for the identification of macrostates and the design of microstates. By applying this theory and method to complex chemical systems, specifically the oil droplet system, I have further enhanced the theory through the utilization of mutual information estimation methods. These methods enable us to determine the number of macrostates present in a given system. The analysis of the macrostate of oil droplets has revealed several intriguing observations. Despite the system having six parameters, its macrostate is only two-dimensional, as evidenced by both the parameter and trajectory perspectives. Furthermore, even with hundreds of dimensions in the trajectory, only 1.8 bits of information can be controlled by the parameters. This low mutual information aligns with the discussion presented in the introduction, wherein observation is regarded as a combination of laws and boundary conditions. A system, governed by laws or rules and influenced by its initial state and environment, exhibits a blend of these two aspects. By identifying quantities that are mutually predictive, we can distinguish between laws and boundary conditions. From this viewpoint, the formulation of the Relational Macrostate Theory can be interpreted differently: the discovery of macrostates entails identifying quantities that are resilient to uncertainty, including external noise and boundary conditions.

The unsupervised architecture of MacroNet offers the advantage of designing microstates without explicitly classifying them, as the identified macrostates serve as continuous labels for the classes. For instance, in Section 3.5.3 of Chapter 3, MacroNet can be utilized for the parameter design of Turing patterns. Additionally, in Chapter 4, it can aid in the chemical compound design of oil droplets. Given the generative capability of MacroNet, it is highly suitable for tasks involving a multitude of unknown information. The relational macrostate theory, coupled with MacroNet, holds substantial promise for applications in both geological and climatic systems. Geological studies often encompass vast amounts of information, much of which can be interpreted as initial or boundary conditions when viewed through the lens of macrostate theory. Such conditions might encapsulate the precise microstates of the early Earth or the thermal fluctuations and stochastic events characteristic of the contemporary Earth and its biosphere. These complexities render predictions at the microstate level a formidable challenge. Nonetheless, the macrostates of these geological systems retain the potential to predict. A notable application emerges in the domain of time series, extending to climatic time series as well. A case in point is the early warning of dynamical change point of dynamical systems, i.e., the tipping points (Drake and Griffen 2010). Over time, the underlying parameters of the dynamical system can also evolve, and crossing some threshold can lead the system to undergo a qualitative transformation, giving rise to another attractor. This significant shift underscores its importance in both the geological and climatic fields. Viewed through the macrostate theory lens, these predictably variable parameters can be defined as macrostates. As introduced earlier, by circumventing the prediction of microstate details, we can distill more predictable insights. Considering the low probability but high consequence nature of tipping points, estimating their probability becomes crucial. Here, the sampling capabilities of MacroNet can be particularly useful. Therefore, both the relational macrostate theory and MacroNet present valuable applications in geology and climate domains, especially given their inherent uncertainties.

From the lens of machine learning and the AI for science domain, the relational macrostate theory furnishes a crucial insight for harnessing machine learning in scientific discoveries, as elaborated in section 1.6 of Chapter 1. Specifically, when the aim is to extract laws and regularities from observational data, the models employed should be inherently generative. The recent surge in the advancement of generative models, most notably the diffusion models (Rombach et al. 2022), signals that the identification of general laws is within our grasp. It's worth noting that while our proposed MacroNet uses normalization flow models, we don't wish to rigidly tether the macrostate theory to any specific architecture. As discussed in this section, many generative and contrastive models seem to be candidates in line with the relational macrostate theory. Thus, we anticipate the broad applicability of the relational macrostate theory across various machine learning domains, particularly those handling uncertainty and the pursuit of discerning regularities.

In considering macrostates from a computational perspective, we are reminded of the concepts introduced in Stephen Wolfram's seminal work, *A New Kind of Science* (Wolfram et al. 2002). Herein, he distinguishes between computationally reducible and irreducible systems. A computationally reducible system permits the circumvention of intermediate states to directly deduce the state at a specified time step. For example, with an ideal simple harmonic oscillator, one can directly compute its state at time T from an initial state (x_0, p_0), bypassing the necessity to iterate through time steps from 1 to T. Such reducibility is critical for extrapolating our understanding of a system across extended temporal durations or larger scales. Conversely, many complex systems, such as certain cellular automata – specifically the rule 110 elementary cellular automata (Cook et al. 2004) and the Game of Life (Gardner 1970) – exhibit computational irreducibility. This characteristic renders the analysis of these systems challenging, as one cannot trivially abstract away microstate details and scale them over expansive space-time dimensions. Yet, the concept of irreducibility may not be absolute, evidenced by the existence of biology and sociology fields. These fields, despite their focus on complex systems and interactions, have formulated theories that sidestep microstate details and eliminate the need for sequential simulation. Seen in this light, macrostates could be defined as the computationally reducible aspects of a system. This means, the computational cost of predicting a macrostate shouldn't rise at the same rate or more rapidly as the size or time scale increases. For example, while the chaotic three-body system lacks an analytic solution and is thereby computationally irreducible, its energy remains constant, shows it is reducible in that context. The relational macrostate theory defines macrostates by the mutual predictability of information within data pairs. This computational viewpoint can be reconciled with this predictive paradigm: macrostates encapsulate those quantities that can be predicted over any (or sufficiently long) temporal intervals with a consistent, minimal computational cost. Nonetheless, this interpretation is most congruent with time-evolving dynamical systems. It requires further studies on the relation between general data pairs (such as rule-pattern pairs) and this computational viewpoint.

For the machine learning aspect, the current MacroNet architecture relies on normalization flow models to regulate the output distribution and perform conditional sampling. However, the limited representational capacity of normalization flow models poses strict constraints. Often, achieving satisfactory performance requires both employing a very deep structure and enduring slow training, resulting in time-consuming applications. Future research in this area could explore alternative neural network approaches, such as combining contrastive learning and diffusion models (Rombach et al. 2022).

Interpretability remains a crucial aspect of scientific investigations. A result that is comprehensible to humans offers two distinct advantages beyond prediction or sampling. Firstly, it allows humans to validate the results and develop theories and methodologies based on them. Secondly, a human-understandable outcome involves human intelligence in the process of discovering underlying laws, leveraging our ability to abstract and make analogies. Attaining these two benefits is challenging, yet there exist several approaches that can contribute to achieving this ambitious objective. For instance, the attention method (Vaswani et al. 2017) can be employed, whereby attention mechanisms highlight the important elements for computing macrostates. Additionally, traditional machine learning techniques such as logistic regression or linear models, as utilized in Chapter 4, can prove valuable due to their simplicity. Many of these methods require flexibility in model selection, not solely limited to normalization flows but encompassing any type of neural network.

Beyond the topics above, the concept of macrostates includes additional profound questions. For example, how do the fundamental units of physics relate to macrostates? Can similar units be identified in complex systems? What criteria dictate the reusability of a macrostate? How are different macrostate variables interconnected?

Such questions could form the foundation of a novel subdomain within complexity research, necessitating interdisciplinary collaboration to amalgamate diverse insights and expertise. Physics has a unique, coherent structure, striving for a physical-level comprehension of complex systems may verge on the overly ambitious. Nevertheless, considering certain concepts from physics can promote the study of complex systems to be more systematic, fostering a more integrated understanding.

REFERENCES

Ahnert, Sebastian Edmund. 2017. "Structural properties of genotype–phenotype maps." *Journal of The Royal Society Interface* 14 (132): 20170275.

Anderson, Philip W. 1972. "More is different: broken symmetry and the nature of the hierarchical structure of science." *Science* 177 (4047): 393–396.

Behrmann, Jens, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. "Invertible residual networks." In *International Conference on Machine Learning,* 573–582. PMLR.

Belghazi, Mohamed Ishmael, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. "Mutual information neural estimation." In *International conference on machine learning,* 531–540. PMLR.

Bond-Taylor, Sam, Adam Leach, Yang Long, and Chris G Willcocks. 2021. "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models." *arXiv preprint arXiv:2103.04922.*

Chan, Bert Wang-Chak. 2018. "Lenia-biology of artificial life." *arXiv preprint arXiv:1812.05433.*

Chen, Boyuan, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. 2021. "Discovering State Variables Hidden in Experimental Data." *arXiv preprint arXiv:2112.10755.*

Chen, Ricky TQ, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. 2019. "Residual flows for invertible generative modeling." *Advances in Neural Information Processing Systems* 32.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. "A simple framework for contrastive learning of visual representations." In *International conference on machine learning,* 1597–1607. PMLR.

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." *Advances in neural information processing systems* 29.

Chen, Xinlei, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. "Improved baselines with momentum contrastive learning." *arXiv preprint arXiv:2003.04297.*

Chen, Xinlei, and Kaiming He. 2021. "Exploring simple siamese representation learning." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 15750–15758.

Cook, Matthew, et al. 2004. "Universality in elementary cellular automata." *Complex systems* 15 (1): 1–40.

Cover, Thomas M, and Joy A Thomas. 1991. "Elements of information theory." *Print ISBN 0-471-06259-6 Online ISBN 0-471-20061-1.*

Cybenko, George. 1989. "Approximation by superpositions of a sigmoidal function." *Mathematics of control, signals and systems* 2 (4): 303–314.

Dinh, Laurent, David Krueger, and Yoshua Bengio. 2014. "Nice: Non-linear independent components estimation." *arXiv preprint arXiv:1410.8516.*

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. 2016. "Density estimation using real nvp." *arXiv preprint arXiv:1605.08803.*

Dowson, DC, and BV Landau. 1982. "The Fréchet distance between multivariate normal distributions." *Journal of multivariate analysis* 12 (3): 450–455.

Dozat, Timothy. 2016. "Incorporating nesterov momentum into adam." *ICLR 2016 workshop submission.*

Drake, John M, and Blaine D Griffen. 2010. "Early warning signals of extinction in deteriorating environments." *Nature* 467 (7314): 456–459.

Gardner, Martin. 1970. "The Fantastic Combinations of Jhon Conway's New Solitaire Game'Life." *Sc. Am.* 223:20–123.

Goldstein, Herbert, Charles Poole, and John Safko. 2002. *Classical mechanics.*

Gömöri, Márton, Balázs Gyenis, and Gábor Hofer-Szabó. 2017. "How do macrostates come about?" In *Making it Formally Explicit: Probability, Causality and Indeterminism,* 213–229. Springer.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT press.

Goodfellow, IJ, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio. 2014. *Generative Adversarial Networks (arXiv: 1406.2661). arXiv.*

Granger, Clive WJ. 1969. "Investigating causal relations by econometric models and cross-spectral methods." *Econometrica: journal of the Econometric Society,* 424–438.

Gray, Peter, and Stephen K Scott. 1984. "Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system A+ 2B→ 3B; B→ C." *Chemical Engineering Science* 39 (6): 1087–1097.

Greenbaum, Anne, and Tim P Chartier. 2012. *Numerical methods: design, analysis, and computer implementation of algorithms.* Princeton University Press.

Greydanus, Samuel, Misko Dzamba, and Jason Yosinski. 2019. "Hamiltonian neural networks." *Advances in neural information processing systems* 32.

Gutierrez, Juan Manuel Parrilla, Trevor Hinkley, James Ward Taylor, Kliment Yanev, and Leroy Cronin. 2014. "Evolution of oil droplets in a chemorobotic platform." *Nature communications* 5 (1): 5571.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. "Momentum contrast for unsupervised visual representation learning." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 9729–9738.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 770–778.

Hemmo, Meir, and Orly R Shenker. 2012. *The road to Maxwell's demon: conceptual foundations of statistical mechanics.* Cambridge University Press.

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. "beta–vae: Learning basic visual concepts with a constrained variational framework." In *International conference on learning representations.*

Hinton, Geoffrey E, and Ruslan R Salakhutdinov. 2006. "Reducing the dimensionality of data with neural networks." *science* 313 (5786): 504–507.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33:6840–6851.

Hoel, Erik P. 2017. "When the map is better than the territory." *Entropy* 19 (5): 188.

Hoel, Erik P, Larissa Albantakis, and Giulio Tononi. 2013. "Quantifying causal emergence shows that macro can beat micro." *Proceedings of the National Academy of Sciences* 110 (49): 19790–19795.

Hu, Hong-Ye, Dian Wu, Yi-Zhuang You, Bruno Olshausen, and Yubei Chen. 2022. "RG-Flow: a hierarchical and explainable flow model based on renormalization group and sparse prior." *Machine Learning: Science and Technology* 3 (3): 035009.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. "Image-to-image translation with conditional adversarial networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 1125–1134.

Jaynes, Edwin T. 1957. "Information theory and statistical mechanics." *Physical review* 106 (4): 620.

Jolliffe, Ian T, and Jorge Cadima. 2016. "Principal component analysis: a review and recent developments." *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374 (2065): 20150202.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, et al. 2021. "Highly accurate protein structure prediction with AlphaFold." *Nature* 596 (7873): 583–589.

Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980.*

Kingma, Diederik P, and Max Welling. 2013. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114.*

Kingma, Durk P, and Prafulla Dhariwal. 2018. "Glow: Generative flow with invertible 1x1 convolutions." *Advances in neural information processing systems* 31.

Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger. 2004. "Estimating mutual information." *Physical review E* 69 (6): 066138.

Krizhevsky, A. 2009. "Learning Multiple Layers of Features from Tiny Images." *Master's thesis, University of Tront.*

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25.

Landau, Lev Davidovich, and Evgenii Mikhailovich Lifshitz. 2013. *Statistical Physics: Volume 5.* Vol. 5. Elsevier.

Lazo, A Verdugo, and P Rathie. 1978. "On the entropy of continuous probability distributions (corresp.)" *IEEE Transactions on Information Theory* 24 (1): 120–122.

Levin, Michael. 2014. "Endogenous bioelectrical networks store non-genetic patterning information during development and regeneration." *The Journal of physiology* 592 (11): 2295–2305.

Liu, Ziming, and Max Tegmark. 2021. "Machine learning conservation laws from trajectories." *Physical Review Letters* 126 (18): 180604.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781.*

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems* 26.

Nalisnick, Eric, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. "Improving document ranking with dual word embeddings." In *Proceedings of the 25th International Conference Companion on World Wide Web,* 83–84.

Noether, Emmy. 1971. "Invariant variation problems." *Transport theory and statistical physics* 1 (3): 186–207.

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

Pathak, Jaideep, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. 2018. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach." *Physical review letters* 120 (2): 024102.

Pattee, Howard Hunt, Joanna Rączaszek-Leonardi, and Howard Hunt Pattee. 2012. "Evolving self-reference: matter, symbols, and semantic closure." *Laws, Language and Life: Howard Pattee's classic papers on the physics of symbols with contemporary commentary,* 211–226.

Pearl, Judea, and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect.* Basic books.

Pearson, John E. 1993. "Complex patterns in a simple system." *Science* 261 (5118): 189–192.

Prechelt, Lutz. 2002. “Early stopping-but when?” In *Neural Networks: Tricks of the trade,* 55–69. Springer.

Riedel, Jürgen, and Hector Zenil. 2015. “Cross-boundary behavioural reprogrammability reveals evidence of pervasive universality.” *arXiv preprint arXiv:1510.01671.*

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. “High-resolution image synthesis with latent diffusion models.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 10684–10695.

Rosenblatt, Frank. 1958. “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review* 65 (6): 386.

Seif, Alireza, Mohammad Hafezi, and Christopher Jarzynski. 2021. “Machine learning the thermodynamic arrow of time.” *Nature Physics* 17 (1): 105–113.

Shalizi, Cosma Rohilla, and Cristopher Moore. 2003. “What is a macrostate? Subjective observations and objective dynamics.” *arXiv preprint cond-mat/0303625.*

Strogatz, Steven, Sara Walker, Julia M Yeomans, Corina Tarnita, Elsa Arcaute, Manlio De Domenico, Oriol Artime, and Kwang-Il Goh. 2022. “Fifty years of ‘More is different’.” *Nature Reviews Physics* 4 (8): 508–510.

Turing, Alan Mathison. 1990. “The chemical basis of morphogenesis.” *Bulletin of mathematical biology* 52 (1): 153–197.

Udrescu, Silviu-Marian, and Max Tegmark. 2020. “AI Feynman: A physics-inspired method for symbolic regression.” *Science Advances* 6 (16): eaay2631.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. “Visualizing data using t-SNE.” *Journal of machine learning research* 9 (11).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. “Attention is all you need.” *Advances in neural information processing systems* 30.

Wang, Ce, Hui Zhai, and Yi-Zhuang You. 2019. “Emergent Schrödinger equation in an introspective machine learning architecture.” *Science Bulletin* 64 (17): 1228–1233.

Wolfram, Stephen, et al. 2002. *A new kind of science.* Vol. 5. Wolfram media Champaign, IL.

Zhang, Yanbo, and Sara Imari Walker. 2022. “A Relational Macrostate Theory Guides Artificial Intelligence to Learn Macro and Design Micro.” *arXiv preprint arXiv:2210.07374.*

Zhao, Shengjia, Jiaming Song, and Stefano Ermon. 2017. “Infovae: Information maximizing variational autoencoders.” *arXiv preprint arXiv:1706.02262.*

APPENDIX A

RELATIONAL MACROSTATE THEORY

A.1 Definitions and Theory

A.1.1 Definitions

A.1.1.1 Equivalence

Two microstates are equivalent if and only if they belong to the same macrostate. Using $\sim$ to represent equivalence, we have $u \sim u' \iff \varphi(u) = \varphi(u')$. Here φ maps microstates to macrostates.

A.1.1.2 Relations

I use the inclusive term relation to include most types of paired variables – for instance, co-occurrence pairs, data-label pairs, or past-future pairs, etc. A set of microstate pairs (u_i, v_i) can be mathematically represented by joint distribution $P(u, v)$. This joint distribution represents the entire *micro-to-micro* relations. Given a microstate u_i, micro-to-micro relation can be defined as a conditional distribution $P(v|u = u_i)$ or $P(v|u_i)$.

Since there are two types of data in the paired datasets, I use $\alpha = \varphi_u(u_i)$ and $\beta = \varphi_v(v_i)$ to represent the macrostates of u_i and v_i respectively. For simplicity, I also use φ to represent either of the mappings from microstates to macrostates when there is no ambiguity.

Given the microstates and their macrostates, I can define the entire *micro-to-macro* relation as $P(u, \beta)$ and $P(\alpha, v)$. And the micro-to-macro relation for a certain microstate, say u_i (or v_j), is represented as conditional distributions:

$$P(\beta|u_i) = \int P(\beta|v)P(v|u_i)\mathrm{d}v \tag{A.1}$$

$$P(\alpha|v_i) = \int P(\alpha|u)P(u|v_i)\mathrm{d}u \tag{A.2}$$

Here, the $P(\beta|v)$ is a probabilistic representation of φ_v, which is a many-to-one mapping since φ_v is a deterministic mapping.

The *macro-to-macro* relation can also be represented as the distribution $P(\alpha, \beta)$.

$$P(\alpha, \beta) = \iint P(\alpha|u)P(\beta|v)P(u, v)\mathrm{d}v\mathrm{d}u \tag{A.3}$$

So, macro-to-macro for certain macrostates can be defined as conditional distributions $P(\beta|\alpha_i)$ and $P(\alpha|\beta_i)$. These definitions of relations are illustrated in Figure 21.

(A) entire relations

$$\begin{tikzcd}
\alpha \arrow[r, dash, "P(\alpha,\beta)"] & \beta \\
u \arrow[u, "\varphi_u"] \arrow[ur, dash, "P(u,\beta)"] \arrow[r, dash, "P(u,v)"] & v \arrow[u, "\varphi_v"]
\end{tikzcd}$$

(B) relations for certain states

$$\begin{tikzcd}
\alpha_i \arrow[r, "P(\beta|\alpha_i)"] & \beta \\
u_i \arrow[u, "\varphi_u"] \arrow[ur, "P(\beta|u_i)"] \arrow[r, "P(v|u_i)"] & v \arrow[u, "\varphi_v"]
\end{tikzcd}$$

Figure 21. The Relations Can Be Represented by Joint Distributions and Conditional Distributions. (A) Joint Distributions Are Used to Represent the Entire Relationship. (B) For a Certain Microstate u_i or Macrostate α_i, Conditional Distributions Can Be Used to Represent Its Relations.

A.1.1.3 Definition of Macrostates

Based on the definitions of relations, macrostates can be defined based on micro-to-macro relations:

Definition S1: macrostate. Two pairs of microstates u_i and u_j (and v_i and v_j) belong to the same macrostate if and only if they have the same micro-to-macro relation:

$$u_i \sim u_j \iff P(\beta|u_i) = P(\beta|u_j) \text{ and} \tag{A.4}$$

$$v_i \sim v_j \iff P(\alpha|v_i) = P(\alpha|v_j) \tag{A.5}$$

The macrostate solutions should be self-consistent. Figure 6A shows a consistent solution, as an example, $u_1 \sim u_2$ because $P(\beta|u_1) = P(\beta|u_2)$, and $v_1 \sim v_2$ because $P(\alpha|v_1) = P(\alpha|v_2)$. However, Figure 6B shows an inconsistent solution: The microstates in red circles are all mapped to the orange macrostate in V and therefore should not be mapped to different macrostates in U because each microstate should belong to only one macrostate. They have the same micro-to-macro relation.

Another solution is that all the microstates are mapped to the same macrostate (see Figure 6C). This kind of solution will not provide any meaningful information about the systems under study.

Therefore, in addition to the definition S1, I propose an information criterion to specify “good macrostates”. That is, given a certain number (or dimension) of macrostates, the information criterion imposes the constraint of maximizing the mutual information $I(\alpha;\beta)$ at the macrostate.

A.1.1.4 From Definition to Optimization Objective

In relational macrostate theory, macrostates are defined in terms of relations. As such they are defined in a circular manner: the macrostate of a microstate is determined by the macrostates of its related microstates. Since the macrostates in U are defined by the macrostates in V, and macrostates in V are defined by U, we need to optimize the mapping from micro-to-macro to find informative and consistent solutions that allow identifying macrostates associated to symmetries. Based on the definition of macrostates, continuation can be applied to the definition by introducing distance functions D_1 and D_2. The continuous version of the definition becomes:

$$D_1[\varphi_u(u_i), \varphi_u(u_j)] = D_2[P(\beta|u_i), P(\beta|u_j)], \tag{A.6}$$

$$D_1[\varphi_v(v_i), \varphi_v(v_j)] = D_2[P(\alpha|v_i), P(\alpha|v_j)]. \tag{A.7}$$

When these two equations are perfectly satisfied, this reduced to the original macrostate definition. When choosing D_1 be square Euclidean distance and D_2 be 2-Wasserstein distance (Dowson and Landau 1982), we can verify the following formula is a solution for our macrostate definition:

$$\varphi_u(u_i) \approx \varphi_v(v_i). \tag{A.8}$$

More specifically, the solution is:

$$\varphi_u(u_i) - \varphi_v(v_i) \sim \mathcal{N}(0, \Sigma), \tag{A.9}$$

where (u_i, v_i) is sampled from $P(u, v)$, and $\mathrm{tr}(\Sigma) \ll 1$. Using $P(\alpha|u_i)$ and $P(\beta|v_i)$ to represent $\varphi_u(u_i)$ and $\varphi_v(v_i)$ as distributions, we have:

$$\begin{aligned} P(\beta|u_i) &= \int_v P(\beta|v)P(v|u_i)\mathrm{d}v && \text{(A.10)} \\ &= \int_v P(\alpha+\delta|u_i)P(v|u_i)\mathrm{d}v && \text{(A.11)} \\ &= P(\alpha+\delta|u_i) && \text{(A.12)} \end{aligned}$$

where $\delta \sim \mathcal{N}(0, \Sigma)$ and $tr(\Sigma) \ll 1$. Here we replaced $P(\beta|u)$ by $P(\alpha + \delta|u_i)$ because $\varphi_u(u_i) \approx \varphi_v(v_i)$, or $\alpha_i \approx \beta_i$. So, we can find that $P(\beta|u_i)$ and $P(\alpha|v_i)$ are both normal distributions with low standard deviations. For normal distributions X and Y, the 2-Wasserstein distance has a simple form:

$$W_2(X, Y)^2 = |\mu_x - \mu_y|^2 + \mathrm{tr}\left(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{1/2}\right), \tag{A.13}$$

So, the definition becomes:

$$|\varphi_u(u_i) - \varphi_u(u_j)|^2 = |\mathbb{E}(\varphi_v(v_i) - \varphi_v(v_j))|^2 + \mathrm{tr}(\Sigma_i + \Sigma_j - 2(\Sigma_i\Sigma_j)^{1/2}), \tag{A.14}$$

$$|\varphi_v(v_i) - \varphi_v(v_j)|^2 = \left|\mathbb{E}(\varphi_u(u_i) - \varphi_u(u_j))\right|^2 + \mathrm{tr}(\Sigma_i' + \Sigma_j' - 2(\Sigma_i{}'\Sigma_j')^{1/2}), \tag{A.15}$$

Since $\Sigma \ll 1$, we can abandon the trace term and remove the expectations:

$$\left|\varphi_u(u_i) - \varphi_u(u_j)\right|^2 \approx \left|\varphi_v(v_i) - \varphi_v(v_j)\right|^2, \tag{A.16}$$

$$\left|\varphi_v(v_i) - \varphi_v(v_j)\right|^2 \approx \left|\varphi_u(u_i) - \varphi_u(u_j)\right|^2, \tag{A.17}$$

The formulas still hold when substitute $\varphi_u(u_i) \approx \varphi_v(v_j)$ into it. So, we can verify that $\varphi_u(u_i) \approx \varphi_v(v_j)$ is a solution for our definition. This solution can be approximated by minimizing the distance between $\varphi_u(u_i)$ and $\varphi_v(v_i)$. There may exist other more general but more complex solutions. However, this simple approach shows good performance in experiments.

A.2 Methods

A.2.1 Invertibility and Distribution Control

Technically, our framework requires two key features in the neural network for learning φ: the ability to perform conditional sampling and ability to control the distribution of its outputs. Fortunately, the invertible neural networks (INNs) cover both features. The invertibility makes conditional sampling possible. And the distribution control feature makes it possible to avoid trivial solutions without a large number of negative samples (in (Xinlei Chen et al. 2020), 65536 negative samples are used).

In a broad definition, the INNs can be classified into two types: flow-based models (Dinh, Krueger, and Bengio 2014; Dinh, Sohl-Dickstein, and Bengio 2016; R. T. Chen et al. 2019), and models that are trained to be invertible such as InfoGAN (Xi Chen et al. 2016). The flow-based model, including the coupling models such as RealNVP (Dinh, Sohl-Dickstein, and Bengio 2016), NICE (Dinh, Krueger, and Bengio 2014), and ResNet-based models such as invertible residual networks (Behrmann et al. 2019) and ResFlow (R. T. Chen et al. 2019). The flow-based models have two common designs: first, they are guaranteed to be invertible, no matter how well they have been trained. Second, they are easy to compute determinants of Jacobians.

With the information of determinants of Jacobians, the probability density of the output can be computed by the “change of variable” theorem (Dinh, Sohl-Dickstein, and Bengio 2016), hence we can control the distribution of output. Here, for simplicity, let's just consider an extreme case: if a linear matrix that maps a three-dimensional manifold to a zero, one, or two-dimensional manifold that is embedded in three-dimensional space. Then, the rank of the matrix must be two or lower. Hence, the determinant of Jacobian will be zero. So, by avoiding having zero determinants of Jacobians, we can avoid the dimension collapse, hence avoiding trivial solutions.

Another type of INNs is the models that are trained to be invertible. Such models should also have the two features as flow-based models: invertibility, and distribution control. InfoGAN (Xi Chen et al. 2016) architecture is an example that follows the requirements. Compared to vanilla GANs, the InfoGAN is simply doing two different things: 1) splitting the input noise into two parts c and z. 2) add a Q network that can reconstruct the c information, i.e., $Q[G(c,z)] \to c$, where G is the generator. The inverse of InfoGAN is trained, it can *partially* inverse the process of $G : (c,z) \to x$ by using $Q : x \to c$, while the z information is lost. This loss will not affect our macrostate framework, because we can map microstates to macrostates by $Q : u \to \alpha$, and sample microstates from macrostates by $G : (\alpha, z) \to u$. The ability of distribution control is achieved by the reconstruction process and discriminator together. Given that discriminator exists, if c is sampled from a distribution P and $z \sim \mathcal{N}(0,1)$, then $G(c,z)$ will follow the data distribution. Since Q is trained to predict c by the generated samples, as an inverse process, $Q(x \sim P_{data})$ will follow the distribution of P. By controlling the distribution, InfoGAN can also avoid trivial solutions.

Our experiments have all been trained on flow-based models. We are making this choice for three reasons: 1) flow-based models are guaranteed to be invertible. and 2) flow-based models are not likely to have mode collapse problems, while GAN based models often have such problems. This is critical if we want to design microstates. 3) flow-based models make the experiments more concise. However, the InfoGAN structure can still be useful when we need a high expressivity because it can use more different neural network structures.

A.2.2 Invertible Neural Networks

Table 3 compares different types of INNs. The forward and inverse column shows the mapping from input x to output y, and y to x.

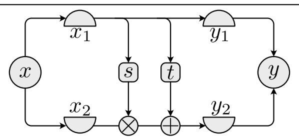
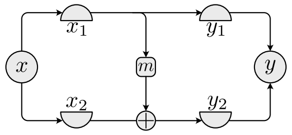
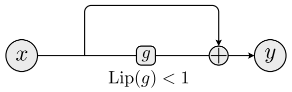
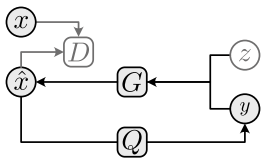
name	structure	forward	inverse
RealNVP	 x_1 y_1 x s t y x_2 y_2	$y_1 = x_1$ $y_2 = x_2 s(x_1) + t(x_1)$	$x_1 = y_1$ $x_2 = (y_2 - t(y_1))/s(y_1)$
NICE		$y_1 = x_1$ $y_2 = x_2 + t(x_1)$	$x_1 = y_1$ $x_2 = y_2 - t(y_1)$
ResFlow		$y = x + g(x)$	$x = \lim_{n\to\infty} x_n$ $x_n = y - g(x_{n-1})$ if Lip$(g) < 1$
InfoGAN		$y = Q(\hat{x})$	$\hat{x} \sim G(y, z)$

Table 3. Illustrations for Multiple Versions of Invertible Neural Networks (INNs).

A.2.2.1 Coarse-graining and Sampling

The flow-based models require the output and input to have the same dimensions for invertibility. So, to do coarse-grain and up sampling, we need to adopt a special way to change dimensions.

(Hu et al. 2022) provided a multi-scale architecture, which let the network abandon dimensions: $f : x \to (y, z)$, where z is the abandoned dimensions, and y can be used to do supervised or self-supervised training. In this way, we can reduce the dimension and do coarse graining. In the forward process, given a N-dimensional input, the output will be splitted into two variables $\alpha^{(D)}$ and $z^{(N-D)}$, where the superscripts show their dimensions. Only α will be trained to satisfy $\varphi_u(u_i) = \varphi_v(v_i)$. To make it clear, we use φ to represent the mapping from u to α, and use Φ to represent the mapping from u to (α, z).

However, z is not totally ignored. Since we also want to do conditional sampling, the distribution of z should also be trained to be an independent normal distribution. So, the Jacobian of φ is computed by Φ so we can include z. When doing conditional sampling, given the macrostate $\alpha^{(D)}$ or $\beta^{(D)}$, we sample a $z^{(N-D)}$ to compute $\Phi^{-1}(\alpha, z)$. The coarse-graining and sampling process are summarized in Table 4.

	forward	**training**
coarse-graining	$\Phi(u^{(N)}) = (\alpha^{(D)}, z^{(N-D)})$, where $z^{(N-D)}$ is the abandoned dimensions. And $\varphi(u) = \alpha$.	both α and z will be trained to follow independent normal distribution. z will *not* be stored since we know its distribution after training. And α will be trained as a macrostate.
sampling	$\varphi^{-1}(\alpha) = \Phi^{-1}(\alpha^{(D)}, z^{(N-D)}) = u^{(N)}$, where $z^{(N-D)}$ is sampled from an independent normal distribution.	no sampling used in training

Table 4. Details of Coarse-graining and Sampling Process.

Since (α, z) is trained to be independent normal distributions, the $P(z|\alpha)$ should also follow normal distribution. With this feature, we are able to do conditional sampling of u from $P(u|\varphi(u) = \alpha)$.

A.2.3 Training Tricks

The flow-based models have limitations of expressivity (Bond-Taylor et al. 2021) since their Jacobian and dimensions are restricted. A common way to overcome this problem is to have more layers of INNs, for example, the Glow model (Kingma and Dhariwal 2018) uses nearly one hundred layers to do generative tasks on the CIFAR10 dataset (Krizhevsky 2009). However, for some tasks which have very low dimensions, more layers cannot provide results that are good enough. To solve this problem, we propose two useful tricks for different situations.

Noisy Kernel Trick

The expressivity problem can often be overcome by adding more layers of INNs (Bond-Taylor et al. 2021). However, our experiments show that when the input dimension is too low, adding layers will not help. While extending neural networks wider can significantly improve the performance. To extend the neural network of INN, we need to extend the input dimension by concatenating the original input with additional random variables:

$$u' = [u, x], x \sim N^d(0, 10^{-3}) \tag{A.18}$$

With this method, we can add d dimensions to the inputs. Here, the u is the original input, and x is the appended input, which is sampled from a standard normal distribution. Note that x has to be sampled from a d-dimensional distribution instead of zeros. This is because the flow-based model will be trained to map inputs to an independent normal distribution. However, if we append inputs by zeros, the input itself will be a lower dimensional manifold, which makes it impossible to be mapped to an independent normal distribution and leads to unstable training. We found that 10^{-3} is a good standard deviation that is small enough to reduce the interference from noise, and large enough to avoid the explosion of log-Jacobian. Since this method is increasing the input dimensions with gaussian noise, we call it "noisy kernel". The additional dimensions will increase the expressivity of flow-based models, which will lead to better performance. Table 5 shows noisy kernels can significantly improve the performance on the simple harmonic oscillator task.

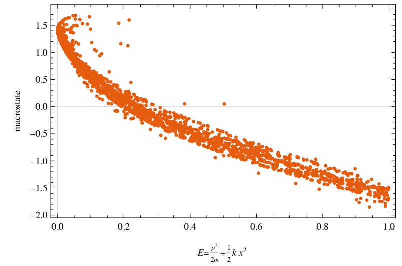
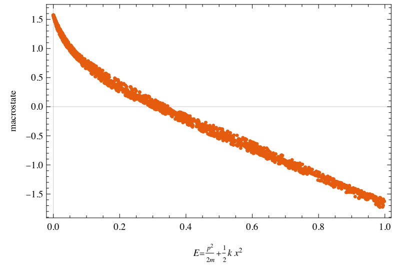
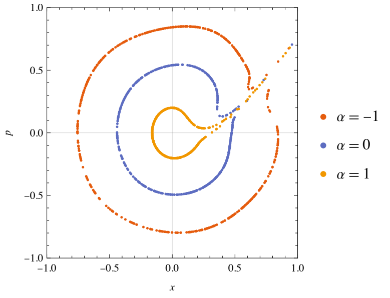
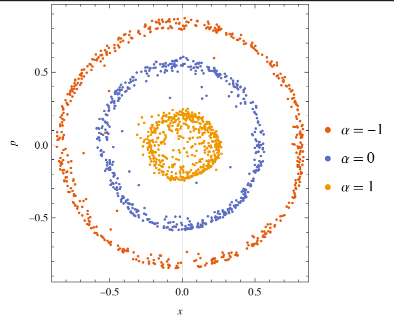
	without noisy kernel	with noisy kernel
A		
B		

Table 5. The Noisy Kernel Can Improve the Performance When the Input Dimension Is Too Low. However, Noisy Kernels May Make the Sampling Noisy.

However, the added noise will also have side effects on sampling. The additional dimensions in z will add noise to the output when doing sampling (see Table 5). So, we only suggest using noisy kernels when necessary, for example, when the dimension is too low.

One-side INN structures

In many cases, only one side of microstates needs to be sampled. In such a case, we only need to let one of two networks (i.e., φ_u or φ_v) be invertible. The other network is not necessary to be an INN. This makes the optimization much easier since the free-form neural networks will have higher expressivity. We adopted this method in finding macrostates of Turing patterns.

Putting batch normalization at the last layer

The common practice in neural networks often puts the linear layer as the last layer. In MacroNet, although we have the distribution term to avoid trivial solutions, we still find that putting the invertible batch normalization layer (Dinh, Sohl-Dickstein, and Bengio 2016) as the last layer (or before the last resize layer) will improve the performance. This may be caused by the potential tradeoff between the prediction loss and the distribution loss, which could skew the distribution of macrostates away from gaussian distribution. This trick cannot omit the importance of the distribution loss. Even when the macrostates have a standard deviation of one, the macrostates can still be low-dimensional manifolds that lack information.

Neural network choosing

While MacroNet is designed to uncover hidden symmetries from observations, the specific choice of neural networks or certain details can influence the outcome. For example, Table 5 illustrates how a noisy kernel can impact the identification of a macrostate. However, in real-world applications, only changes in symmetry typically result in significantly different outcomes. We employed convolutional neural networks (CNNs) in our Turing pattern experiments, but not in linear dynamical systems. This decision was driven by the need to process images derived from Turing patterns. Since many of these images inherently possess translational symmetry, CNNs are a more suitable choice. Though there's a need to make manual selections regarding network specifics, knowledge from the machine learning community offers practical guidance on network design.

A.3 Experiments

A.3.1 Linear Dynamical Systems

A linear dynamical system can be represented as a differential equation:

$$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = M\vec{x} \tag{A.19}$$

where M is a $n \times n$ matrix. n is the dimension of vector $\vec{x}$. So, when the system

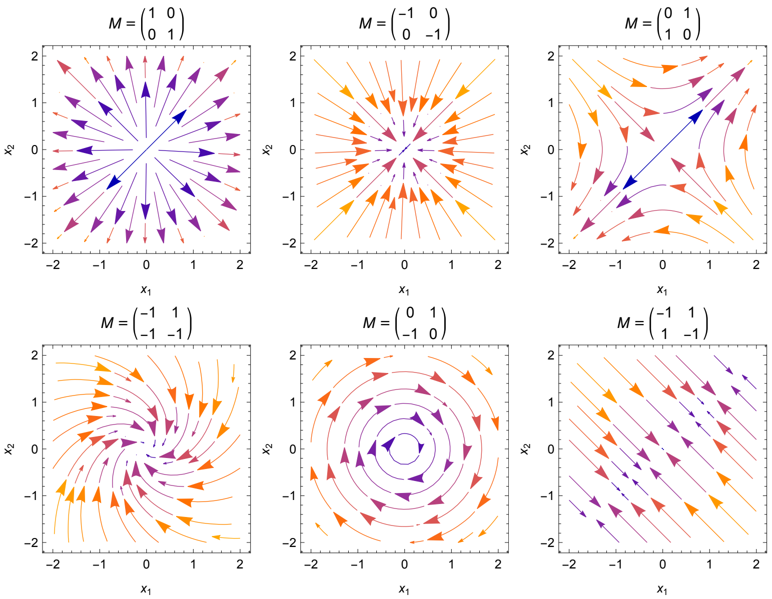


Figure 22. The Behavior of the Linear Dynamical System Is Changing with the Matrix M.

has different $\vec{x}$, the $\mathrm{d}\vec{x}/\mathrm{d}t$ will be different. Different matrices will lead to different behaviors, such as attractor, limit cycles, rotations or saddles (see Figure 22).

So, there exist many-to-many mappings between the matrix and trajectory:

1. one-to-many: For the same matrix M, depending on initial states, the trajectories can be different. For instance, given $M = I$, the trajectories can move to the right or left if the initial state $x_0 = (1, 0)$ or $(-1, 0)$.
2. many-to-one: Also, even with different matrices, the trajectories can be the same when the initial state is properly chosen. For instance, when $M_1 = I$, and M_2 be a permutation matrix that permutes between dimension 1 and 2, their trajectories can be the same when the initial state $x_0 = (\xi, \xi)$, where $\xi > 0$.

For such many-to-many mapping situations, our macrostate theory and machine learning method can help us design the matrices for given trajectories. Here we define the macrostates on the parameter-trajectory pairs. The parameter is a 2×2 matrix M and the trajectory is a $n \times 2$ tensor $x_{0:n-1} = [x_0, x_1, ..., x_{n-1}]$ to represent the evolution with the initial state x_0, where $n = 8$. We coarse-grained both sides to a 2-dimensional space as the macrostate (see Figure 23).

The training data is generated by an algorithm. For each $(u = M, v = x_{0:n-1})$ pair, the M is firstly sampled from an independent normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$.

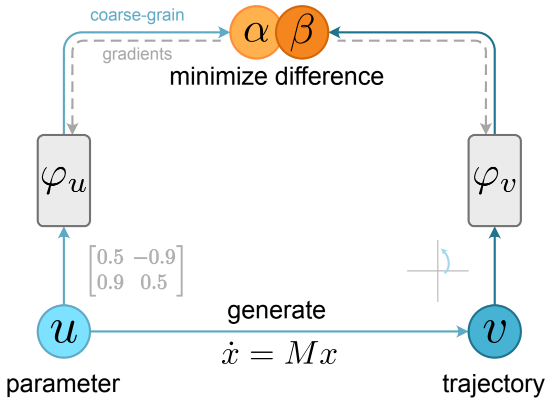


Figure 23. The Training Process of Finding Macrostates from Linear Dynamical Systems. Both the Parameters and Trajectories Are Coarse-grained to Two-dimensional Macrostates.

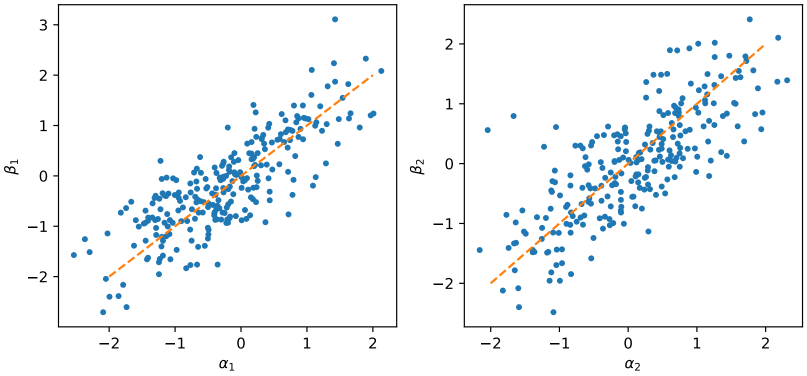


Figure 24. The Trained Neural Network on Linear Dynamic Systems Is Capable of Predicting the Macrostate. Here, Each Point Represents a (α_i, β_i) Pair. Here, $\alpha_i = \varphi_u(u_i)$ and $\beta_i = \varphi_v(v_i)$. By Jointly Compare α_i and β_j, We Can Quantify the Performance of the Predictions on Macrostates. In the View of Mutual Information, When All Points Are on the Curve $\beta = \alpha$, the Mutual Information $I(\alpha; \beta)$ Is Maximized.

Then the trajectory is generated by the dynamic $\mathrm{d}x/\mathrm{d}t = Mx$, where the initial state x_0 are sampled uniformly and independently in a 2-dimensional space $U^2(-1, 1)$.

The training takes 2000 epochs, and each epoch has 512 samples with a batch size of 256. We use Adam optimizer (Kingma and Ba 2014) to train the model. The learning rate is 10^{-3} and the weight decay is 10^{-5}. We let $\gamma = 0.1$ to balance the prediction loss and distribution loss. Figure 24 shows the scatter plot of macrostates (α, β), which indicates the accuracy of prediction at macrostates.

After training, we can do two things: given a trajectory s_e as "example behavior", use φ_v^{-1} to sample other trajectories that have the same macrostate as s_e. Or, given a trajectory, use φ_u^{-1} to sample parameters that can generate this trajectory with certain initial states. Here we show the sampling with different example behaviors (represented by $x_{0:n-1}$, illustrated by red trajectories) in Figure 25.

Neural network architecture

The neural network maps the parameters and trajectories to a two-dimensional space as the macrostates. To improve the performance, we use noisy kernels to improve the performance. For the parameter side, we use a noisy kernel to increase the dimension from 4 to 8. For the trajectory side, we use a noisy kernel to increase the dimension from 16 to 32. The noises for each additional dimension are independently sampled from $\mathcal{N}(0, 10^{-3})$. The details of the structure of the neural networks are in Table 6. The one-dimensional INN block is composed of a linear INN (Kingma and Dhariwal 2018), a RealNVP 1-dimensional layer, and an invertible batch normalization layer (Dinh, Sohl-Dickstein, and Bengio 2016).

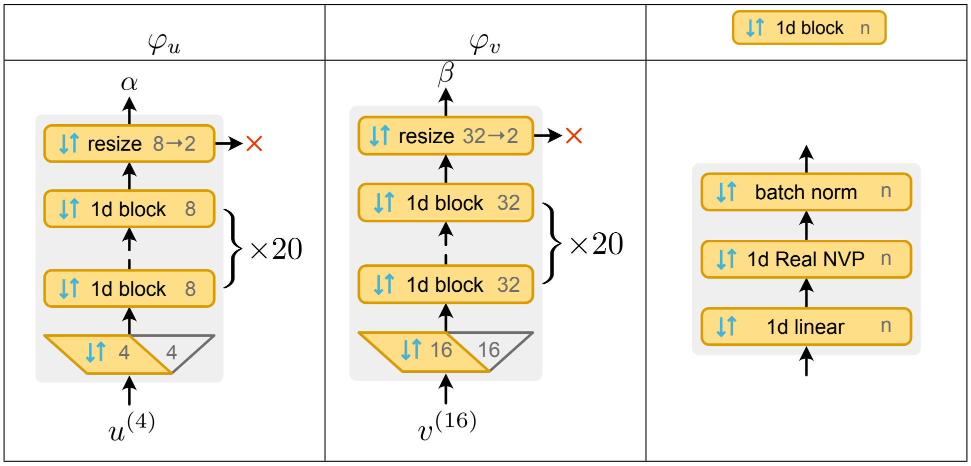


Table 6. Here We Adopted the Noisy Kernels, Represented by Trapezoids. For φ_u, the 4-Dimensional Microstates Input Are Increased to 8 Dimensions by the Noisy Kernels. After That, There Are 20 One-Dimensional INN Blocks (Indicated by the ⇃↾ Icon). At the End, We Simply Abandon 6 Dimensions to Get a 2-Dimensional Output.

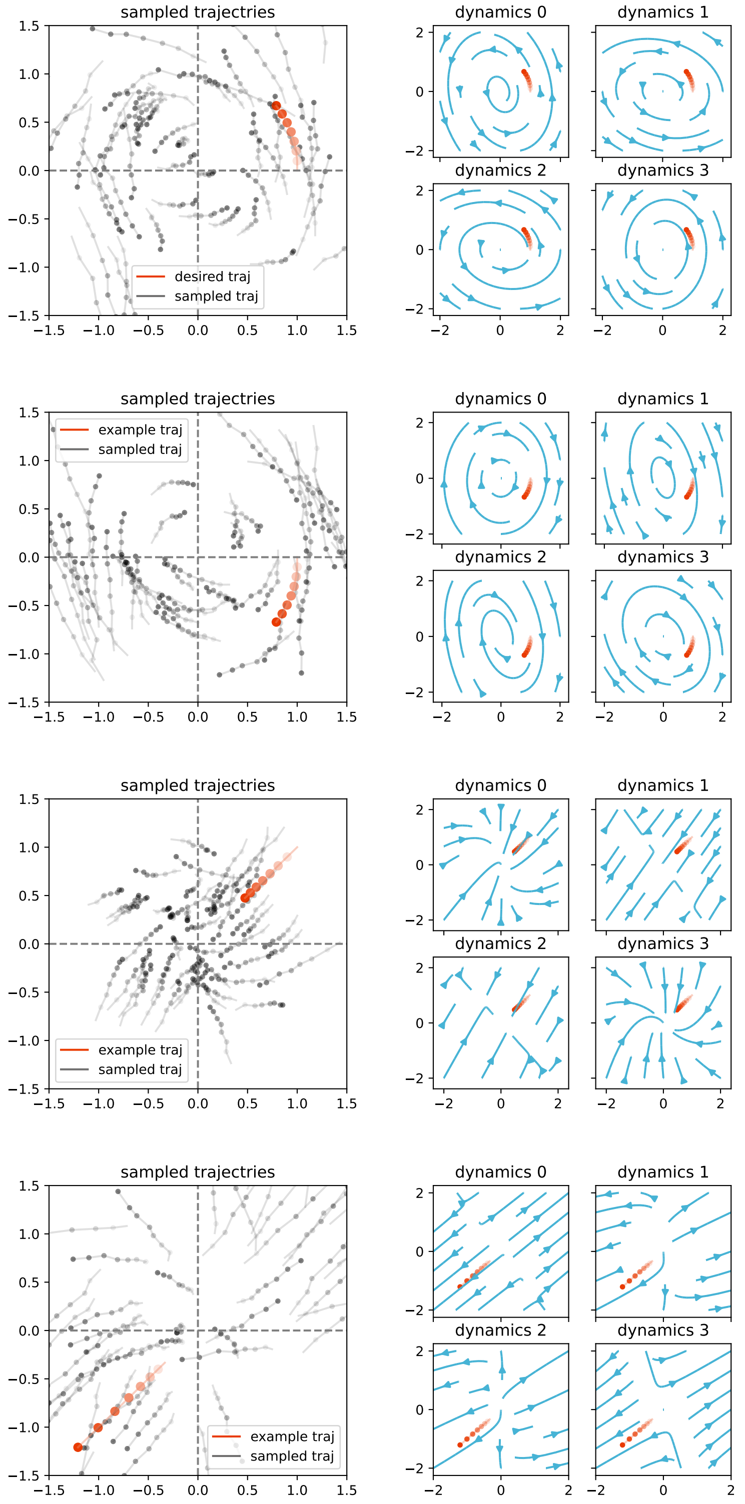


Figure 25. Red Lines Show the Example Trajectories. And Gray Dotted Lines Show the Sampled Microstates of Trajectories. The Blue Vector Lines Represent the Dynamics of Sampled Parameters.

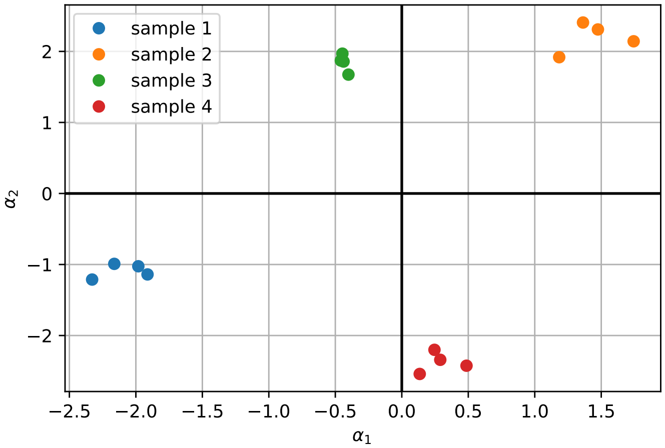


Figure 26. By Feeding the Sampled Matrices From Figure 25 Into φ_u, We Can Compute Their Macrostates. Samples 1 and 2 Differ in Their Rotation Directions, While Samples 3 and 4 Differ in Their Radial Directions. They Exhibit Distinct Embeddings in the Macrostate Space.

A.3.2 Simple Harmonic Oscillator

There is an important special case of the macrostates. When the relation is built on temporally connected microstates, the neural network is predicting future macrostates, which is similar to the contrastive predictive learning (Oord, Li, and Vinyals 2018), but adding the conditional sampling ability. Furthermore, if we force the two neural networks to share the same parameter, then it is learning time invariant quantities. Here we use simple harmonic oscillators (SHOs) as an example. The Hamiltonian of SHOs is:

$$H(x,p) = \frac{p^2}{2m} + \frac{1}{2}kx^2, \tag{A.20}$$

where $p = mv$ is the momentum, x is the position, m is the mass, and k represents the elasticity of the spring. In this experiment, we let $m = 1$ and $k = 1$ in all cases for simplicity. So, the solution is:

$$x_t = A\cos(t+\phi),\ \ p_t = -A\sin(t+\phi), \tag{A.21}$$

where A depends on the initial energy, $A = \sqrt{x_0^2 + p_0^2}$. And ϕ is the initial phase, $\phi = \arctan(p_0/x_0)$. The microstate of simple harmonic oscillator is (x_t, p_t). To find an invariant quantity, we require the macrostate of $u = (x_0, p_0)$ should as close as the macrostate of $v = (x_\tau, p_\tau)$, where τ follows the uniform distribution $\mathcal{U}(0, 2\pi)$ (shown

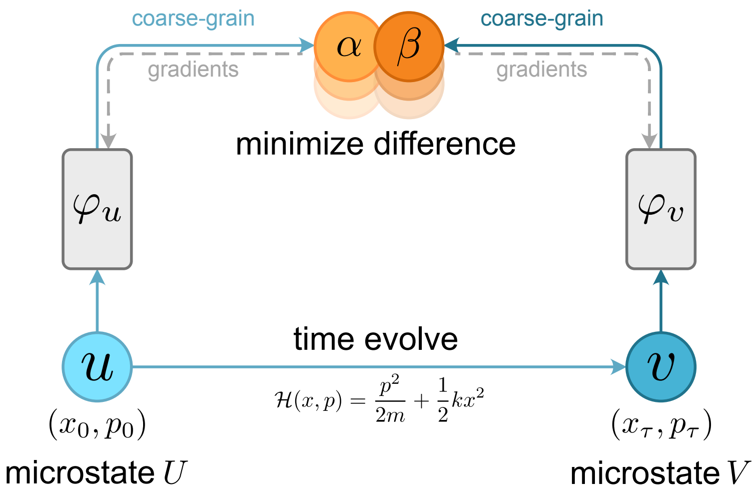


Figure 27. The Training Process of Finding Invariants as Macrostates from Simple Harmonic Oscillators.

in Figure 28A). Since τ is a random variable, predicting microstate (x_τ, p_τ) is not possible. However, the macrostate can be predictable. The training architecture is shown in Figure 27.

We use 2048 samples of (u, v) pairs to train the neural network. The training takes 200 epochs with a batch size of 256. We use NAdam optimizer (Dozat 2016) to optimize the neural network. The learning rate is 5×10^{-3}. The learning rate decreases by 0.1 in each 60 epochs. To balance the prediction loss and distribution loss, we choose $\gamma = 0.5$.

Figure 28B shows the invariant quantity found by our neural network has a clear and monotonous relation to the energy. We can also sample the microstates (x, p) from given invariant by implement $\varphi^{-1}(\alpha)$. The results show that the neural network can sample a ring in (x, p) space (Figure 28C), which is exactly the solution of $p^2 + x^2 = H$.

Neural network architecture

Since the dimension of the microstate is two, we use a noisy kernel to increase it to eight dimensions. The noise follows the distribution of $\mathcal{N}^6(0, 10^{-3})$. We also use residual flow (R. T. Chen et al. 2019) as the basic block to increase the expressivity. The details of the neural network are shown in Table 7. Note that here we let φ_v shares the same weight as φ_u.

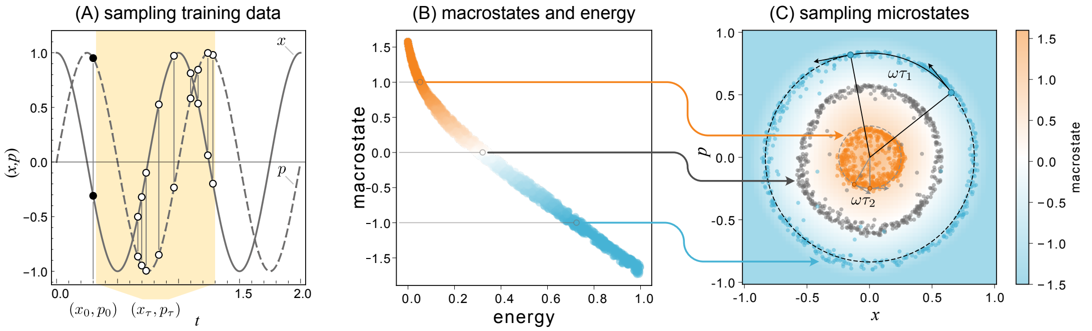


Figure 28. With a Simple Harmonic Oscillator, We Train a Neural Network to Find Invariant Quantities as a Special Case of Macrostates. **(A)** The (u, v) Pairs Are Sampled from Simulations, Where $u = (x_0, p_0)$ (the Black Dots) and $v = (x_\tau, p_\tau)$. The τ Is Sampled from a Uniform Distribution $\mathcal{U}(0, 2\pi)$. The White Dots in the Yellow Region Show a Sampling Example of v. Due to the Randomness of τ, It Is Impossible for Accurate Prediction at Microstate. **(B)** The Neural Network Learns Energy as the Invariant Quantity. The x-Axis Is the Energy of Microstates Computed by the Physical Theory of SHOs Discovered by Humans, and the y-Axis Is the Macrostate Discovered by the Neural Network. They Show a Monotonical Relation, Which Implies the Successful Identification of Energy by the Neural Network. **(C)** Conditional Sampling Microstates from $P((x,p)|\varphi(x,p) = \alpha_i)$, Where the α_i Are the Given Macrostates. The Results Approximate Equal Energy Surfaces, Denoted by the Dashed Circles. Note That the Noise in the Sampling Is a Side Effect of the Noisy Kernel Trick We Use Here. The Background Color Also Shows the Learned Macrostate Mapping as a Field.

A.3.3 Turing Patterns

The Turing patterns are two-dimensional patterns generated by reaction-diffusion models (Turing 1990). By changing the parameter of the model, the reaction-diffusion model can generate many different types of patterns (Pearson 1993). In this experiment, we use macrostate theory to find the macrostate of the patterns and parameters. Then, we sample parameters that can generate certain types of patterns.

Here we use the Gray-Scott model (Gray and Scott 1984) as the reaction-diffusion model. In this model, there are two types of chemical components, their densities are represented as the density fields a and b. The dynamics is represented by the following differential equations:

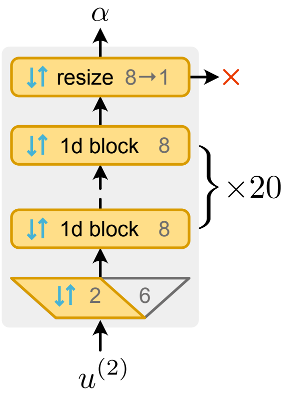
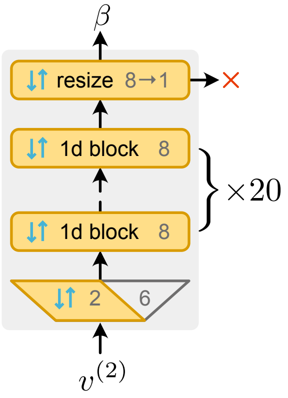
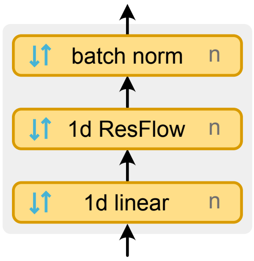
φ_u	$\varphi_v = \varphi_u$	1d block n
		

Table 7. The Neural Network Structure for Finding Macrostates of Simple Harmonic Oscillators. To Find Invariant Quantity, the φ_u and φ_v Have the Same Structure and Share the Same Weights. We Replaced the RealNVP Layer with the ResFlow Layer to Get Better Performance.

$$\frac{\partial a}{\partial t} = D_a \nabla^2 a - ab^2 + F(1-a), \tag{A.22}$$

$$\frac{\partial b}{\partial t} = D_b \nabla^2 b + ab^2 - (F+k)b, \tag{A.23}$$

where D_a, D_b, F and k are four positive parameters that determine the behavior of the system. So, a microstate u here is a vector of the four parameters, i.e., $u = (D_a, D_b, F, k)$. And the microstate v is the pattern generated based on the parameters. When initializing the a, b as 64×64 grids, each elements are independently sampled from the uniform distribution $\mathcal{U}(0,1)$. We approximate the differential equation on a $2 \times 64 \times 64$ tensor by using Euler method (Greenbaum and Chartier 2012) with step size $\mathrm{d}t = 0.1$.

We only sample (u, v) pairs that have meaningful structure in the v matrix and omit the cases where v is a blank image (all elements in v have the same value) with no structure. Using this method, we sample 1024 pairs of microstates. The training architecture is shown in Figure 29.

We trained the neural network 1000 epochs with NAdam optimizer. The learning rate is 10^{-3}. To help the training converge, we reduce the learning rate by 0.5 every 128 epochs. To balance the prediction loss and distribution loss, we let $\gamma = 0.1$.

Since we do not want to sample the pattern v, we only let φ_u be invertible, and let φ_v be a free form neural network. This will make φ_v has higher expressivity and

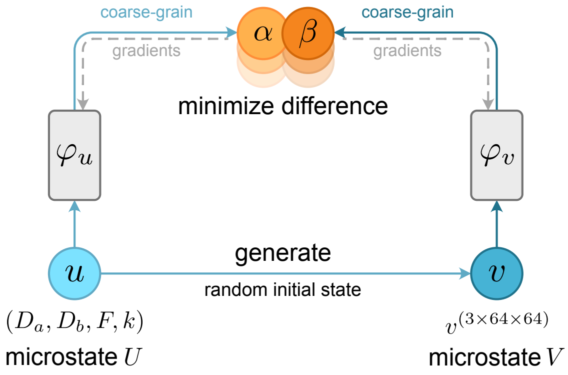


Figure 29. The Training Process of Finding Macrostates from Turing Patterns. The Neural Networks Maps the Parameter (D_a, D_b, F, k) and Patterns $v^{(3\times 64\times 64)}$ to Macrostates in a Two-Dimensional Space.

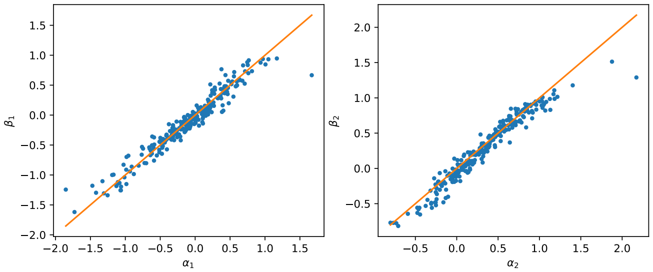


Figure 30. The Trained Neural Network on Turing Patterns Is Capable of Predicting the Macrostate. Here, Each Point Represents a (α_i, β_i) Pair. Here, $\alpha_i = \varphi_u(u_i)$ and $\beta_i = \varphi_v(v_i)$. Here We Map the Microstates to a Two-Dimensional Space, so We Compare the Macrostates on Each Dimension, Represented as α_i and β_i.

easier to be optimized. The φ_u uses 5 invertible blocks and one resize block to reduce the dimension from 4 to 2. Each invertible block contains an invertible linear layer, a Real-NVP layer, and a batch normalization layer. The φ_v is a convolutional neural network that maps $3 \times 64 \times 64$ tensor to a two dimensional vector. Note that the channel is changed from 2 to 3 by the mapping $(a, b) \to (a, b, (a + b)/2)$ to make it have better visualization and easier to do data augmentations, while not losing or alter any information. The detailed neural network structure is shown in Table 8.

Figure 30 compares the macrostates mapped from parameters (α) and macrostates mapped from patterns (β). Most points are laying on the $\alpha = \beta$ line, which indicates that this trained neural network made good predictions at macrostate and having high mutual information $I(\alpha; \beta)$.

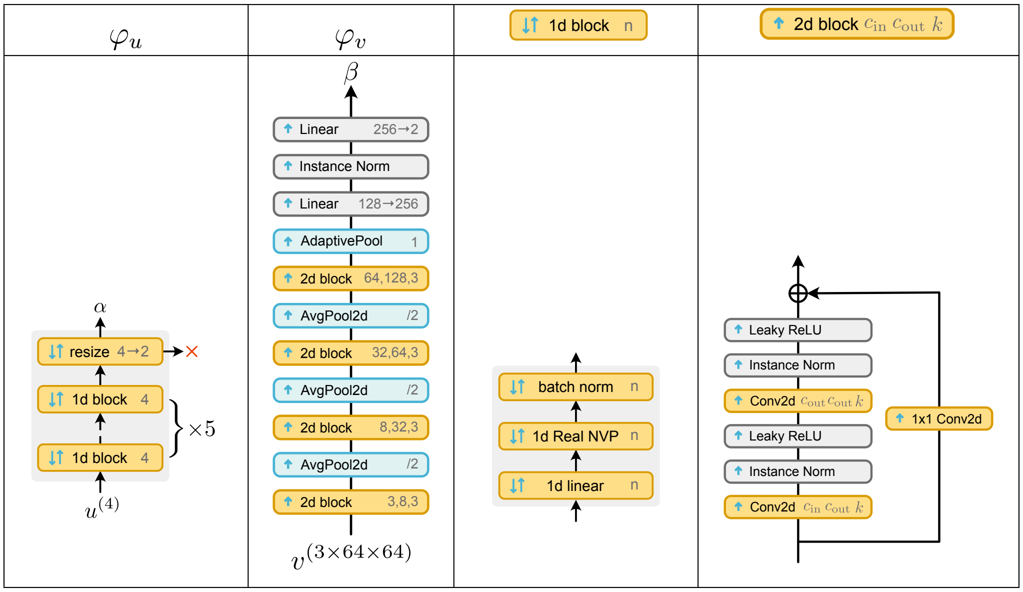


Table 8. The Neural Network Structure for Finding Macrostates of Turing Patterns. For the Parameter Side (φ_u) We Use a 5-Layer INN to Get a 2-Dimensional Output. For the Pattern Side (φ_v), Since Generation Is Not Needed, We Use a Free-Form Neural Network to Get a 2-Dimensional Output.

APPENDIX B

UNCOVERING GENOTYPE-PHENOTYPE MAPPING IN COMPLEX CHEMICAL SYSTEM BY IDENTIFYING MACROSTATES

B.1 Sampling the Most Likely Microstate

For empirical validation, we choose chemical parameters from four macrostates. Rather than sampling the distribution, our approach was to sample the most likely microstate given certain macrostates, thereby identifying the most effective parameters. The MacroNet used in this study is built on normalization flow models, which include variants such as NICE, Real-NVP, ResFlow, Glow, and more. A key advantage of these models is their ability to compute the probability density of the input with relatively low computational cost. Normalization flow models estimate the probability density by utilizing the following change of variable formula:

$$\log(p_U(u)) = \log(p_A(f(u))) + \log\left(\left|\det \frac{\partial f(u)}{\partial u}\right|\right) \tag{B.1}$$

In this equation, u represents the input microstate, $f(u)$ the output, and p_A the prior distribution. The second term is the log determinant Jacobian. Due to the existence of the second term, computing the most likely microstate (microstates with highest probability density at normalized microstate space) becomes non-trivial. To compute the most likely microstate with a given macrostate, we sample 2048 microstates. Then, since normalization flow model can estimate probability density of input with low cost, we choose the top-n microstates with highest log probability. It is important to mention the normalization of microstates in our process. To enhance the training outcomes, we preprocessed the microstates to approximate normal distributions closely. For the chemical compound parameters, given their initial uniform distribution, we applied the inverse error function (erf^{-1}) to transform them into a normal distribution. A variable x, following a standard uniform distribution, can be converted into a standard normal distribution via the following equation:

$$y = \sqrt{2}\mathrm{erf}^{-1}(2x - 1). \tag{B.2}$$

Ideally, the log determinant Jacobian term should encompass this preprocessing function to pinpoint the most likely microstate in the original parameter space. Nevertheless, we chose to identify the most likely microstate in the preprocessed parameter space, rather than the original parameter space, for two main reasons. Firstly, the inverse error function has singular points at -1 and 1. Owing to the unavoidable imperfections of neural networks, these points could lead to inaccurate density estimations. Secondly, as the inverse error function is a monotonous function, the results in the two spaces will not differ qualitatively. For these reasons, we are sampling microstates based on their probability density in the normalized chemical parameter space.