ACMP: Allen-Cahn Message Passing for Graph Neural Networks with Particle Phase Transition

Yuelin Wang[*]
Shanghai Jiao Tong University

Kai Yi[*]
UNSW

Xinliang Liu[†]
KAUST

Yu Guang Wang[†]
Shanghai Jiao Tong University

Shi Jin[†]
Shanghai Jiao Tong University

Abstract

Neural message passing is a basic feature extraction unit for graph-structured data that takes account of the impact of neighboring node features in network propagation from one layer to the next. We model such process by an interacting particle system with attractive and repulsive forces and the Allen-Cahn force arising in the modeling of phase transition. The system is a reaction-diffusion process which can separate particles to different clusters. This induces an Allen-Cahn message passing (ACMP) for graph neural networks where the numerical iteration for the solution constitutes the message passing propagation and GNN prediction that enables node classification due to the formation of multi-clusters, helped by the phase transition of particles. ACMP can propel the network depth to hundreds of layers with theoretically proven strictly positive lower bound of the Dirichlet energy and the formation of multiple clusters. It thus provides a deep model of GNNs which circumvents the common GNN problem of oversmoothing. Experiments for various real node classification datasets, with possible high homophily difficulty, show the GNNs with ACMP can achieve state of the art performance with no decay of Dirichlet energy.

1 Introduction

Graph neural networks (GNNs) have gained a champion and received a great attention in the past five years due to its powerful expressiveness and approximation capability for learning graph structured data and broad applications [6, 8, 14, 15, 31, 68]. Neural message passing [33] has served as a fundamental feature extraction unit for graph-structured data that aggregates the features of neighbors in network propagation. Many graph message passing propagators suffer from oversmoothing when node features are indistinguishable as network depth goes high. PDEs such as diffusion equations [17] and coupled oscillator system [40] have been used to define graph convolution that enable to avoid oversmoothing, where each iteration of the numerical solution to the differential equation is regarded as a neural message passing update and the iteration number as the depth of network layers. We develop a message passing propagator based on Allen-Cahn particle evolution in which phase transition clusters particles into desired flocks. The proposed scheme preserves the Dirichlet energy in network propagation and circumvents oversmoothing when the network depth is high.

The behavior of particle system based message passing is similar to that of collective behaviors common in nature and human society [25, 11, 41, 65]. For example, insects form swarms to work; birds forms flocks to immigrate; humans forms parties to express public opinions. Various

[*]equal contribution
[†]corresponding author

Preprint. Under review.

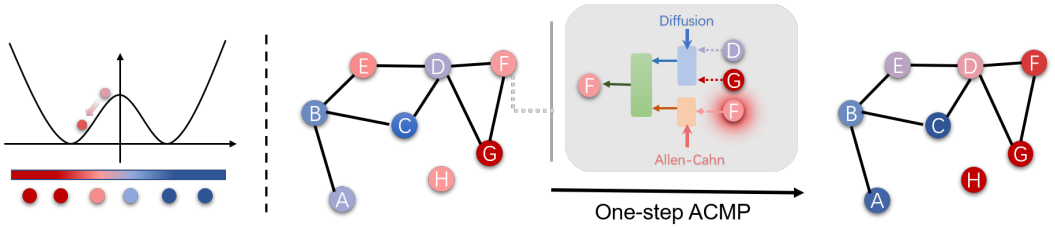

Figure 1: The left figure shows the double-well potential field $W(x)$ and particles $\{x_i\}$ according to their position gains their own potential energy (with different colors). In the Allen-Cahn dynamics, nodes with shallow blue or shallow red color have the tendency to move toward dark blue or dark red particles. The right figure illustrates the one-step ACMP propagator. Isolated node H has no interaction force on it, which then tends into dark red. The remaining nodes are influenced by both the potential field and the interplay with their neighbors. Take node F for example. Suppose all the interaction forces are attractive. The node F is driven by a strong 'red attraction' from G and a weak 'purple attraction', besides its own red traction. F will then turn into red in the next layer.

mathematical models have been proposed [3, 49, 16, 59, 24] deriving from statistical physics [2, 44], sociometry [48, 47] to interpret these behaviors. Many of them are agent-based models that use interacting particle systems. Such particle equations can be properly designed to allow the particle evolution to form multi-clusters or consensus depending on the system's intrinsic nature. Since many common topological structures in graphs emerge from self-organization behaviors of multi-agents, a graph has high similarity with a particle system [52]. This motivates us to design a particle system based neural message passing propagator for graph neural networks, where nodes are particles and edges represent the interactions of particles. Each node in the network interacts with their neighbors and the whole network becomes a community to produce either one consensus or several main clusters. The solution at a specific time t can serve as the message passing at layer t.

Our model is based on an interacting particle system with attractive/repulsive force and a force induced by the Allen-Cahin phase transition model. It can be viewed as the graph version of the bi-cluster Cucker-Smale swarming model introduced in [29]. This natrually induces a GNN model to simulate the message passing dynamics. There are two major components in this model. First, the co-existence of the attractive and repulsive forces allows particles to separate into two clusters. The Allen-Cahn [4] or Rayleigh friction [61] term, used to describe phase transitions in physics, prevents the Dirichlet energy in the evolution from becoming unbounded. More importantly, the Allen-Cahn term forces most of particles to stay near its two stable yet distinct local equilibria (two phases). These force terms provide the key mechanism to overcome the oversmoothing problem of deep networks. Specifically we will prove that under suitable conditions in the parameters, the dynamics of the particle system will time-asymptotically form two different clusters and the Dirichlet energy has a strictly positive lower bound.

The proposed model which we call Allen-Cahn Message Passing (ACMP) is featured by three key points: 1) it has the separation power for graph nodes due to the repulsive force; 2) the Dirichlet energy of the GNNs with ACMP remains bounded, due to introducing the Allen-Cahn double-well potential term in the particle equation; 3) the adaption of the proposed GNNs for homophilic and heterogeneous node classification tasks can be adjusted flexibly by training. The model can then reach an acceptable trade-off on self-features and neighbor effect.

2 The Allen-Cahn Particle System

Message Passing in Graph Neural Networks Graph neural networks are a kind of deep neural network which takes graph data as input. Neural Message Passing (MP) [33, 9] is a most widely used propagator for node feature update in GNNs, which takes the following form: with $\mathbf{x}_i^{(k-1)} \in \mathbb{R}^d$ denoting node features of node i in layer $(k-1)$ and $a_{j,i} \in \mathbb{R}^D$ edge features from node j to node i,

$$\mathbf{x}_i^{(k)} = \gamma^{(k)}\left(\mathbf{x}_i^{(k-1)}, \square_{j\in\mathcal{N}_i}\, \phi^{(k)}\left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, a_{j,i}\right)\right),$$

where $\square$ denotes a differentiable, (node) permutation invariant function, e.g., sum, mean or max, and γ and ϕ denote differentiable functions such as MLPs (MultiLayer Perceptrons), and $\mathcal{N}_i$ is the set of

one-hop neighbors of node i. The message passing updates the feature of each node by aggregating the self-feature with neighbors' features. Many GNN feature extraction modules such as GCN [39], GAT [66] and GIN [70] can be written as message passing. For example, the MP of GCNs reads, with learnable parameters $\mathbf{\Theta}$,

$$\mathbf{x}_i' = \mathbf{\Theta}^\top \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{a_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j. \tag{1}$$

Graph attention network (GAT) uses attention coefficients $\alpha_{i,j}$ as similarity information between nodes in the MP update

$$\mathbf{x}_i' = \alpha_{i,i} \mathbf{\Theta} \mathbf{x}_i + \sum_{j \in \mathcal{N}_i} \alpha_{i,j} \mathbf{\Theta} \mathbf{x}_j, \tag{2}$$

with

$$\alpha_{i,j} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_k]\right)\right)}. \tag{3}$$

The MP framework was also developed as PDE solvers in [13] by embedding differential equations as a parameter into message passing like [12]. This paper regards particle PDE evolution as message passing propagation, and appropriate design of the particle system offers desired properties for the resulting GNN.

Interacting particle systems The interior self-organized system can be viewed as a coarse message passing network [33, 14]. Each particle moves/changes as a reaction of aggregated information from its neighbours and itself. Such process becomes an analogy to the judgement process for certain unknown signals in classification or prediction tasks. Oversmoothing is a critical difficulty the usual GNN models, such as GCN [39] and GAT [66] cannot work well [54, 56] when features become too smooth to be used for class identification and the Dirichlet energy has exponential decay. In the context of particle systems, this means that all the particles (i.e. node features) reach a consensus or produce a mono-cluster flocking asymptotically. By contrast, equilibria of multi-clusters give rise to separable features naturally.

The Allen-Cahn equation for phase transitions To guarantee separated clusters in our model, we add an extra potential to the dynamics inspired by the modelling of phase transition, which is one of the most interesting aspects of many particle systems [67]. Phase transition is the transformation of a substance from one state to another under certain conditions of temperature and pressure [1]. For example, imposing proper pressure on a bottle of nitrogen, some of the gas will change into liquid. Ice melting to water, water boiling to steam, graphite transforming into diamond are all common phase transforms in real world. For a 'binary mixture' of one material in two or more phases, the proportion of the various phases will change in some velocity and finally achieve the equilibria. The Allen-Cahn equation is a reaction-diffusion process including order-disorder transitions: $u_t = \mu^2 \Delta u + u(1-u)(1+u)$ for $\mu > 0$, which, in long time, pushes any initial data to two stable equilibria (phases) ± 1.

2.1 The Allen-Cahn Energy

It is a natural extension of the continuous partial differential equation model to discrete graph model. The variational principle governing many PDE models states that the equilibrium state is actually the minimizer of one specific energy. The equilibrium state carries meaningful information and can therefore be used as embedded features in the context of machine learning. We first introduce the *Dirichlet energy* and show that GCNs can be characterized by looking into the corresponding Euler-Lagrange equation of the Dirichlet energy. Based on this, we introduce the *Allen-Cahn energy* and identify the Allen-Cahn Message Passing simultaneously.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $|\mathcal{V}| = N$ nodes and $|\mathcal{E}| = E$ edges, and $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature defined on the i^{th} node and set $\mathbf{x} = \oplus \mathbf{x}_i$. Let adjacent matrix $\mathbf{A}$ represent the undirected connectivity between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, with $a_{i,j} = 1$ for $(i,j) \in \mathcal{E}$ and $a_{i,j} = 0$ for $(i,j) \notin \mathcal{E}$. Let $\mathcal{N}_i : \{j \in \mathbb{N} : a_{i,j} \neq 0\}$ denote the index set of i^{th} node's neighbors.

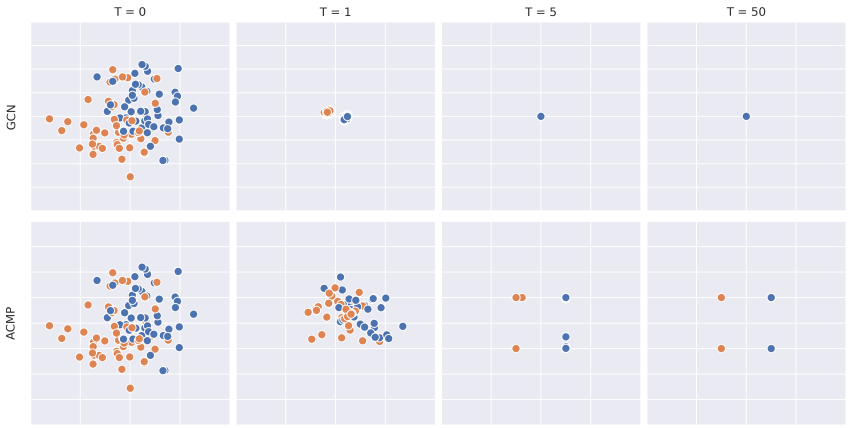

Figure 2: We compare the evolution of node features in GCN and ACMP. We show GCN in the first row and ACMP in the second row. The initial position is represented by the 2-dimensional position of the nodes, which is shown in the first column. The GCN aggregates all node features by taking the weighted average of its neighbors's features. With the propagated steps increasing, all the nodes' features shrink to a point, which gives rise to oversmoothing. When it comes to ACMP, nodes' features are grouped by four attractors, which helps to circumvent oversmoothing.

The Dirichlet energy and its relation with GCNs The Dirichlet energy E in terms of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and node features $\mathbf{x} \in \mathbb{R}^{N \times d}$ takes the form

$$\mathbf{E}(\mathbf{x}) = \frac{1}{N} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} a_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2. \tag{4}$$

Then, the graph neural diffusion scheme [17] can be characterized by considering the time-dependent graph diffusion problem. By calculus of variation, we can formulate the particle equation

$$\frac{\partial \mathbf{x}}{\partial t} = -\nabla_{\mathbf{x}} \mathbf{E}, \quad \frac{\partial \mathbf{x}_i}{\partial t} = -\frac{\partial \mathbf{E}}{\partial \mathbf{x}_i} = \frac{2}{N} \sum_{j \in \mathcal{N}_i} a_{i,j} (\mathbf{x}_j - \mathbf{x}_i). \tag{5}$$

Here, the node features $\mathbf{x}_i$ is a function of t, which is the continuous counterpart of the layers in traditional GCN networks. On the RHS of (5), the summation takes over the one-hop neighbors $\mathcal{N}_i$ of node i, which aggregates the impact from the neighboring nodes.

The graph Allen-Cahn equation Similarly, we define the *Allen-Cahn energy* on graph $\mathcal{G}$ denoted by $\Phi : L^2(\mathcal{V}) \to \mathbb{R}$, as a combination of the Dirichlet energy and double-well potential $W : \mathbb{R}^d \to \mathbb{R}_+$, $W(\mathbf{x}_i) = (\delta/4)(1 - \|\mathbf{x}_i\|^2)^2$, $\Phi(\mathbf{x}) = \frac{1}{2}\alpha \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} a_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 + \sum_{i \in \mathcal{V}} W(\mathbf{x}_i)$, with parameters $\alpha, \delta > 0$ to balance two types of energy. Using this combined energy, we then write the Allen-Cahn equation on the graph $\mathcal{G}$ as

$$\frac{\partial \mathbf{x}}{\partial t} = -\nabla_{\mathbf{x}} \Phi, \quad \frac{\partial \mathbf{x}_i}{\partial t} = -\frac{\partial \Phi}{\partial \mathbf{x}_i} = \alpha \sum_{j \in \mathcal{N}_i} a_{i,j} (\mathbf{x}_j - \mathbf{x}_i) + \delta \mathbf{x}_i (1 - \|\mathbf{x}_i\|^2). \tag{6}$$

The Dirichlet energy acts as the source of attractive force (message passing in the context of GNN), which smooths out the difference between connected node features $\mathbf{x}_i$ over time since the particles eventually reach consensus [35, 50]. This phenomenon has been frequently observed in GNN models. The Allen-Cahn potential will push the particles toward the local equilibrium points of the double-well potential. See the illustration in Figure 1. We use this properties of the equation in the design of our message passing which we introduce now.

3 The Allen-Cahn Message Passing

In the context of message passing on graphs, we use the Allen-Cahn potential energy to balance the node features and edge features in a systematic way as the Dirichlet energy encourages message

passing and interplay between nodes while the Allen-Cahn potential energy function imposes the features' tendency of staying at the same potential well. The competition of these two types of energy determine how node features $\mathbf{x}$ evolve collectively.

As shown in Figure 1, the $\mathbf{x}_i$ is initialized at $t = 0$, and the Allen-Cahn potential encourages the node feature $\mathbf{x}_i$ to fall into the bottom of one well in the evolution. Meanwhile, $\mathbf{x}_i$ is also attracted by other $\mathbf{x}_j$ via edge $(i, j) \in \mathcal{E}$.

We propose the *Allen-Cahn Message Passing* (ACMP) neural network based on equation (6), which learns the node features evolution. Besides the trainable filter kernel, rather than the original deterministic equation, the ACMP assigns learnable coefficients to balance the Dirichlet energy and Allen-Cahn potential.Specifically, the ACMP reads

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha} \odot \sum_{j\in\mathcal{N}_i} a(\mathbf{x}_i(t), \mathbf{x}_j(t))(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \boldsymbol{\delta} \odot \mathbf{x}_i(t) \odot (1 - \mathbf{x}_i(t) \odot \mathbf{x}_i(t)). \tag{7}$$

Here $\boldsymbol{\alpha}, \boldsymbol{\delta} \in \mathbb{R}^d$ are learnable vectors of the same length as the node feature $\mathbf{x}_i$. All terms are channle-wise operations for d channels, except $a(\mathbf{x}_i(t), \mathbf{x}_j(t))$, and $\odot$ represents channel-wise multiplication for d feature channels. We treat the Allen-Cahn potential channel-wisely. This would introduce varying collective behaviors between channels, so that the Allen-Cahn process can be rich enough to separate nodes, as illustrated by Figure 2.

Attractive and Repulsive Force Moreover, we can add learnable repel force on edges to enhance the collective behavior of the ACMP propagation. A repel force is important for learning heterophilic datasets, where the connected nodes are very likely to fall apart into different classes [29]. Our model utilizes the learned repulsive force to make it possible for connected nodes to fall apart into different wells. In section 4, we will also discuss repulsive force which plays an important role in easing the oversmoothing problem on graph. The interplay between nodes i, j are determined by the learnable function $a(\mathbf{x}_i(t), \mathbf{x}_j(t)) \in \mathbb{R}$. We derive two different types of ACMP in terms of how to represent attractive and repulsive forces based on (7).

ACMP-GCN:

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha} \odot \sum_{j\in\mathcal{N}_i} (a_{i,j}^{\mathrm{GCN}} - \beta)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \boldsymbol{\delta} \odot \mathbf{x}_i(t) \odot (1 - \mathbf{x}_i(t) \odot \mathbf{x}_i(t)), \tag{8}$$

where $a_{i,j}^{\mathrm{GCN}}$ is the normalized GCN adjacent matrix, which can be calculated by those used in graph convolutions. For example, using coefficients in GCNs in (1), $a_{i,j}^{\mathrm{GCN}} := a_{i,j}/\sqrt{\hat{d}_i\hat{d}_j}$ is the normalized adjacent matrix $\hat{A}$ with $\mathrm{diag}(\hat{d}_1, \ldots, \hat{d}_N)$ the degree matrix for $\hat{A}$;

ACMP-GAT: we can replace $a_{i,j}^{\mathrm{GCN}}$ in (8) by the attention coefficients (3) of GAT, which contain extra trainable parameters to measure the similarity information between two nodes.

Here, $\beta \in \mathbb{R}^+ \cup \{0\}$ in (8) is a hyperparameter allowing i^{th} and j^{th} nodes to repel each other when $a_{i,j}^{\mathrm{GCN}} - \beta < 0$ or $a^{\mathrm{attn}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - \beta < 0$ respectively. One can adjust β such that the attractive force and repulsive force both present in the graph to enrich the message passing effect. To our best knowledge, this is the *first time* to introduce a type of message passing to amplify the difference between connected nodes by learnable repulsive force (message passing). Note that, if we choose β such that the collective repulsive force exceeds a certain level, the node features $\mathbf{x}$ will blow up over time. The Allen-Cahn potential offers a remedy by simply adjusting the weights parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$. The stability of our model will be discussed in Section 4. Besides, if one chooses $\boldsymbol{\delta} = 0, \beta = 0$ in ACMP-GCN or ACMP-GAT in (8) in [17], our model is reduced to the graph neural diffusion network (GRAND). In experiments, we would make significant use of nontrivial $\boldsymbol{\delta}$ and β.
Do we need to mention $\beta_{i,j}$?

Adding the repulsive force and the Allen-Cahn term in ACMP provides distinct features not shared by previous model [17]. The repulsive forces help the particles to separate into different clusters. The Allen-Cahn term helps the separated clusters to stay near its local equilibia ± 1, rather than be attracted to other clusters. These mechanisms thus help to prevent oversmoothing. Also, the Allen-Cahn term only depends on the feature of the i^{th} node itself. It then links with can be viewed as a metamorphosis of the residual, which enables to enhance the representation of the node's own feature.

4 The Dirichlet Energy and Phase Transition

In order to overcome the oversmoothing phenomenon [54, 56], one needs to study the emergent behaviours of our model. The analysis of emergent behaviours of the neural ODEs for ACMP is usually done through the estimate of the Dirichlet energy. Oversmoothing phenomenon means all node features converge to the same constant – consensus forms – as the network deepens, thus the Dirichlet energy will decay to zero time-asymptotically. Therefore, in principle, if only in one channel the node feature fails to become identical to another one, the oversmmothing will not appear. Recall the Dirichlet energy (4). It suffices to prove that the Dirichlet energy has a strictly positive lower bound. In our model, the node features in each channel tend to evolve into two clusters departing from each other under certain conditions, and the Dirichlet energy has a strictly positive lower bound, as will be shown in this section.

Definition 1 (Oversmoothing) *Let $\mathbf{x}^l$ denotes the hidden features of the l^{th} layer of an L-layer GNN, with $l = 0, \dots, L$. oversmoothing is defined as the exponential convergence to zero of the layer-wise Dirichlet energy as a function of l, i.e., $\mathbf{E}(\mathbf{x}^l) \leq C_1 \exp(-C_2 l)$, with some constants $C_1, C_2 > 0$.*

If $\delta = 0, \beta = 0$, there are only attractive forces in the system, or more generally in any particle system $\{\mathbf{x}_i\}$ satisfying $\dot{\mathbf{x}}_i = \sum_j a_{i,j}(\mathbf{x}_j - \mathbf{x}_i)$ for $a_{i,j} \geq 0$. One can prove the convex hull of $\mathbf{x}(t)$ will not dilate in time. The proof can be found in [50]. Under appropriate assumptions, (e.g. $\mathbf{A}$ being positive), the convex hull readily shrinks into one point. In other words, all the features tend to the same constant, causing the oversmoothing problem. We also include the evolution of the above system in terms of Dirichlet energy here.

Propsition 1 *Let $\mathbf{D}$ denote the degree matrix, i.e., $\mathbf{D} := \mathrm{diag}(d_1, \cdots, d_N)$, where $d_i = \sum_j a_{i,j}$, $a_{i,j} = a_{j,i} \geq 0$. Then $\mathbf{D} - \mathbf{A}$ is symmetric positive semi-definite with the eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{\max} < \infty$. Let $\lambda_{\min} > 0$ be the smallest positive eigenvalue, then for all $t \geq 0$, there exits a constant c such that $\mathbf{E}(\mathbf{x}(t)) \leq c \exp(-\lambda_{\min}^2 t)$.*

However, by adding $\beta > 0$ and the Allen-Cahn term, the attractive force between connected nodes is no longer the only force affecting the system. The force turns to repulsive when $a_{i,j}^{\mathrm{GCN}}$ or $a^{\mathrm{attn}}(\mathbf{x}_i, \mathbf{x}_j)$ is less than β. Depending on the relative strength between $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$, one can achieve a trade-off in the dynamics. Generally, the Dirichlet energy of (8) can be bounded in time thanks to the Allen-Cahn term [34, 29]. Under proper initialization, one can expect separation phenomenon after sufficiently long time. We provide some propositions here confirming this assertion, under certain conditions. We put all the proofs and some supplementary related results in the Appendix.

Propsition 2 *The node features $\mathbf{x}_i$ in ACMP-GCN (8) or ACMP-GAT is bounded for all $t > 0$ if $\delta > 0$, i.e., there exists $R > 0$ such that $\|\mathbf{x}_i\|_\infty \leq R$ for each node i.*

Propsition 3 *If $\delta > 0$, the node features $\mathbf{x}_i$ in ACMP-GCN (8) or ACMP-GAT is bounded in terms of $\|\cdot\|$ and energy for all $t > 0$, i.e., $\mathbf{E}(\mathbf{x}(t)) \leq C$, and $\|\mathbf{x}\| \leq C$, where the constant C only depends on N and $\lambda_{\max}$.*

The lower bound below shows that the Dirichlet energy will never decay to zero. We facilitate the analysis by emergent behavior analysis like those done in [29] (see Appendix for details). In the $\beta = 0$ case, then intuitively one can only expect one cluster since there is no repulsive force between nodes pushing particles away from each other [34]. For a graph $\mathcal{G}$ with N nodes, its vertices are said to form *bi-cluster flocking* if there exist two disjoint sets of vertex subsets $\{\mathbf{x}_i^{(1)}\}_{i=1}^{N_1}$ and $\{\mathbf{x}_j^{(2)}\}_{j=1}^{N_2}$, and two cluster centers c_1, c_2, $|c_1 - c_2| > c > 0$ such that $|\mathbf{x}_i^{(1)} - c_1| < \epsilon_1$ for any i and $|\mathbf{x}_j^{(2)} - c_2| < \epsilon_2$ for any j, and $c > \epsilon_1 + \epsilon_2$. We defined a different bi-cluster flocking in the Appendix. $c > \epsilon_1 + \epsilon_2$ seems unnecessary for "energy lower bouned'.

We now show the long-time behaviour of model (8) following the analysis of [29] for strength coupling (α, δ) that satisfies the following condition: there exists $\{\beta_{i,j}\}$ such that $\mathcal{I} := \{1, \dots, N\}$ can be divided into two disjoint groups $\mathcal{I}_1, \mathcal{I}_2$ with N_1 and N_2 particles respectively:

$$\begin{aligned} 0 < S \leq \overline{a}_{i,j} \quad &\text{with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \text{for } i,j \in I_1, \\ 0 < S \leq \overline{a}_{i,j} \quad &\text{with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \text{for } i,j \in I_2, \\ 0 \leq \overline{a}_{i,j} \leq D \quad &\text{with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \text{otherwise}, \end{aligned} \tag{9}$$

where S, D are independent of time t. In ACMP-GCN, $(a_{i,j}^{\mathrm{GCN}} - \beta_{i,j})$ are parameters and we omit the superscript GCN in $\overline{a}_{i,j}(t)$. For time $t \geq 0$, suppose $\mathbf{x}_c^{(1)}(t)$ and $\mathbf{x}_c^{(2)}(t)$ are the 'feature centers' of the two groups of the particles $\{\mathbf{x}_i^{(1)}(t)\}_{i=1}^{N_1}$ and $\{\mathbf{x}_j^{(2)}(t)\}_{j=1}^{N_2}$ which are partitioned as above from the whole vertex set $\mathcal{V}$, given by

$$\mathbf{x}_c^{(1)}(t) := \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{x}_i^{(1)}(t), \quad \mathbf{x}_c^{(2)}(t) := \frac{1}{N_2}\sum_{i=1}^{N_2}\mathbf{x}_i^{(2)}(t).$$

Suppose $\mathbf{x}_c^{(s)}(t)$ has the d-dimensional feature, and let $x_{c,k}^{(s)}(t)$, $k = 1, \ldots, d$, be the k-th (dimension) component of the feature $\mathbf{x}_c^{(s)}(t)$, $s = 1, 2$.

Propsition 4 *The system (8) has a bi-cluster flocking if for each $k = 1, \ldots, d$, the initial $|x_{c,k}^{(1)}(0) - x_{c,k}^{(2)}(0)| \gg 1$, and if there exists a positive constant η such that*

$$\alpha(S - D)\min\{N_1, N_2\} \geq \delta + \eta.$$

The lowercase x means $\mathbf{x}_k$ for some channel k. So the notation $x_{c,k}$ may be confusing.

Propsition 5 *For system (8) with bi-cluster flocking, there exists a constant $C > 0$ and some time T^* such that $\forall t \geq T^*$,*

$$|\mathbf{x}_i^{(1)}(t) - \mathbf{x}_j^{(2)}(t)| \geq C > 0, \quad \forall i, j.$$

Thus, if the non-zero $a_{i,j}$ are all positive, the Dirichlet energy for ACMP is lower bounded by a positive constant.

5 Implementation and Model Variants

Architecture Suppose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has N nodes and d-dimensional node-wise features represented by a matrix $\mathbf{x}^{\mathrm{in}}$ where row i represents feature of node i. Our scheme first embeds the node feature $\mathbf{x}(0) = \mathrm{MLP}(\mathbf{x}^{\mathrm{in}})$ by a simple multi-layer perceptron, which is treated as an input for the ACMP propagation $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$, by $\mathbf{x}(0) \mapsto \mathbf{x}(T)$, where

$$\mathbf{x}(T) = \mathbf{x}(0) + \int_0^T \frac{\partial \mathbf{x}(t)}{\partial t} dt, \quad \mathbf{x}(0) = \mathrm{MLP}\left(\mathbf{x}^{\mathrm{in}}\right),$$

where $\frac{\partial \mathbf{X}(t)}{\partial t}$ is estimated by ACMP defined on $\mathcal{G}$ based on (6). The node features $\mathbf{x}(T)$ at the ending time are feed into an MLP based classifier.

Neural ODE Solver Our method uses an ODE integrator in the network for numerically implementing the proposed ACMP. To obtain the node features $\mathbf{x}(T)$, we need a stable numerical integrator for solving the ODE efficiently and backpropagation of gradients. One could use explicit and implicit numerical schemes such as explicit Euler, 4th order Runge-Kutta method, midpoint method and Dormand-Prince5 method [19, 42, 53, 17]. Since our model is stable in terms of evolution time and Allen-Cahn double-well potential is infinitely differentiable, most numerical schemes work well for our models as long as the step size τ is small enough. Among these methods, we find that Dormand-Prince5 is fast and stable. We leave the details about numerical solver to Appendix.

Computational Complexity The computational complexity of the one-step ACMP is $\mathcal{O}(NEdn_t)$, where n_t, N, E and d are the number of time steps in time interval $[0, T]$, the number of nodes, the number of edges and the number of feature dimension, respectively. Since our model only considers nearest (one-hop) neighbors, E is significantly smaller than that of graph rewiring method [32, 5] and multi-hop method [72].

More clusters We can simply replace the double well potential W by a multi-well potential to generate more equilibria. We provide two alternatives here. One can use a higher-order polynomial to construct additional wells. In general, a $(2k+1)^{th}$ order polynomial can produce $k+1$ stable equilibria in a proper form, which gives rise to more stable clusters. One can also use $\sin((\frac{3}{2} + l)\pi x + \frac{\pi}{2})$, $l = 0, \cdots, k$, defined on the interval $[-1, 1]$ as the multi-well potential, which has $l + 2$ stable equilibria.

Stronger trapping force As the consensus state (i.e. $x_i = x_j$ for all i, j) might not be a global equilibrium of (8), particles could escape from one well of the potential of W to another well. We can circumvent this instablility by enhancing the attraction of the wells, which can be achieved by reducing the diffusion power around wells:

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha} \odot \sum_{j \in \mathcal{N}_i} (a^{\mathrm{GNN}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - \beta)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \left(1 - \mathbf{x}_i(t)^{\odot 2}\right)^{\odot 2} + \boldsymbol{\delta} \odot \mathbf{x}_i(t) \odot \left(1 - \mathbf{x}_i(t)^{\odot 2}\right). \tag{10}$$

where 'GNN' in a^{GNN} can be GCN or attn, and $z^{\odot 2}$ is $z \odot z$. With this modification in (10), in any channel k, if any particle $\mathbf{x}_i^{(k)}$ gets caught in one potential well, then it is not likily to escape:

Propsition 6 *For (10), there exists a proper $\delta' > 0$ such that $x_i^{(k)} \in [-1, -1 + \delta') \cup (1 - \delta', 1]$, then particle $x_i^{(k)}$ cannot transition into another well.*

6 Experiments

6.1 The Dirichlet Energy

We first illustrate the evolution of the Dirichlet energy of ACMP by a undirected synthetic random graph. The synthetic graph has 100 nodes with two classes and 2D feature which is sampled from the normal distribution with the same standard deviation $\sigma = 2$ and two means $\mu_1 = -0.5$, $\mu_2 = 0.5$. The nodes are connected randomly with probability $p = 0.9$ if they are in the same class, otherwise nodes in different classes are connected with probability $p = 0.1$. We compare the performance of GNN models with four message passing propagators: GCNs [39], GAT [66], GRAND [17] and ACMP-GCN. In Figure 3, we show the Dirichlet energy of each layer's output in logarithm scales. It shows that traditional GNNs such as GCNs and GAT suffer oversmoothing as the Dirichlet energy exponentially decays to zero in the first ten layers. Graph Neural Diffusion (GRAND) relieves this problem by multiplying a small constant which can delay all nodes' feature to collapse to the same value. For ACMP, the energy stabilizes at the level that relies upon the roots of the Allen-Cahn polynomial in (7) after slightly decaying in the first two layers.

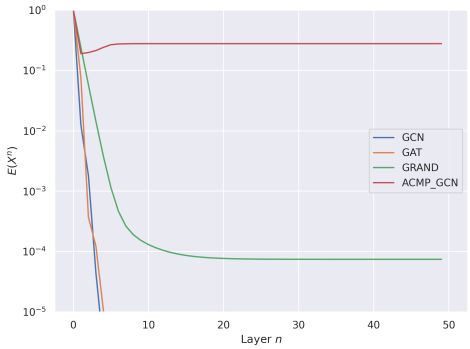

Figure 3: Evolution of Dirichlet energy $\mathbf{E}(\mathbf{X}^n)$ of layer-wise node features $\mathbf{X}^n$ propagated through GCN, GAT, GRAND and ACMP-GCN.

Homophily level	**Texas** 0.11	**Wisconsin** 0.21	**Cornell** 0.30
GPRGNN [20]	78.4 ± 4.4	82.9 ± 4.2	80.3 ± 8.1
H2GCN [72]	84.9 ± 7.2	87.7 ± 5.0	82.7 ± 5.3
GCNII [18]	77.6 ± 3.8	80.4 ± 3.4	77.9 ± 3.8
Geom-GCN [57]	66.8 ± 2.7	64.5 ± 3.7	60.5 ± 3.7
PairNorm [71]	60.3 ± 4.3	48.4 ± 6.1	58.9 ± 3.2
GraphSAGE [36]	82.4 ± 6.1	81.2 ± 5.6	76.0 ± 5.0
MLP	80.8 ± 4.8	85.3 ± 3.3	81.9 ± 6.4
GAT [66]	52.2 ± 6.6	49.4 ± 4.1	61.9 ± 5.1
GCN [39]	55.1 ± 5.2	51.8 ± 3.1	60.5 ± 5.3
GraphCON [40]	85.4 ± 4.2	87.8 ± 3.3	84.3 ± 4.8
ACMP-GCN (ours)	86.2 ± 0.3	86.1 ± 0.4	85.4 ± 0.7

Table 1: Node classification results on heterophilic datasets. We use the 10 fixed split train, validation and test from [57] and show the mean and standard deviation of test accuracy. We show the best three method in red (First), blue (Second), and violet (Third).

6.2 Node Classification

We compare the performance of ACMP with several popular GNN model architectures on various node classification benchmarks, containing both homophilic and heterophilic dataset. Graph data is

considered as *homophilic* if its homophily level [57] is big. In this case, similar nodes in the graph tend to connect together. Conversely, the graph data is said *heterophilic* if it has a small homophily level, when most neighbors do not have the same label with source nodes. We aim to demonstrate that ACMP is a flexible GNN model which can learn well both kinds of datasets by balancing the diffusion and Allen-Cahn terms.

Homophilic datasets The results of our study are presented for the most widely used citation networks: Cora [45], Citeseer [63] and Pubmed [51]. Additionally, we also evaluate our model on the Amazon co-purchasing graphs Computer and Photo [51], and CoauthorCS [64]. We compare our model with traditional GNN models: Graph Convolutional Network (GCN) [39], Graph Attention Network (GAT) [66], Mixture Model Networks [46] and GraphSage [36]. We also compare our results with recent ODE-based GNNs, Continuous Graph Neural Networks (CGNN) [69], Graph Neural Ordinary Differential Equations (GDE) [58] and Graph Neural Diffusion (GRAND) [17]. To address the limitations of this evaluation methodology proposed by [64], we report results for all datasets using 100 random splits with 10 random initializations and show the node classification result with mean and standard deviation in Table 2.

Random Split Homophily level	**Cora** **0.83**	**CiteSeer** **0.71**	**PubMed** **0.79**	**Coauthor CS** **0.80**	**Computer** **0.77**	**Photo** **0.83**
GCN [39]	81.5 ± 1.3	71.9 ± 1.9	77.8 ± 2.9	91.1 ± 0.5	82.6 ± 2.4	91.2 ± 1.2
GAT [66]	81.8 ± 1.3	71.4 ± 1.9	78.7 ± 2.3	90.5 ± 0.6	78.0	85.7
GAT-ppr [66]	81.6 ± 0.3	68.5 ± 0.2	76.7 ± 0.3	91.3 ± 0.1	$\mathbf{85.4 \pm 0.1}$	90.9 ± 0.3
MoNet [46]	81.3 ± 1.3	71.2 ± 2.0	78.6 ± 2.3	90.8 ± 0.6	83.5 ± 2.2	91.2 ± 2.3
GraphSage-mean [36]	79.2 ± 7.7	71.6 ± 2.0	77.4 ± 2.2	91.3 ± 2.8	82.4 ± 1.8	91.4 ± 1.3
GraphSage-max [36]	76.6 ± 1.9	67.5 ± 2.3	76.1 ± 2.3	85.0 ± 1.1	N/A	90.4 ± 1.3
CGNN [69]	81.4 ± 1.6	66.9 ± 1.8	66.6 ± 4.4	$\mathbf{92.3 \pm 0.2}$	80.29 ± 2.0	91.39 ± 1.5
GDE [58]	78.7 ± 2.2	71.8 ± 1.1	73.9 ± 3.7	91.6 ± 0.1	81.9 ± 0.6	$\mathbf{92.4 \pm 2.0}$
GRAND-l [17]	$\mathbf{83.6 \pm 1.0}$	$\mathbf{73.4 \pm 0.5}$	$\mathbf{78.8 \pm 1.7}$	$\mathbf{92.9 \pm 0.4}$	$\mathbf{83.7 \pm 1.2}$	$\mathbf{92.3 \pm 0.9}$
ACMP-GCN (ours)	$\mathbf{84.9 \pm 0.6}$	$\mathbf{75.0 \pm 1.0}$	$\mathbf{78.9 \pm 1.0}$	$\mathbf{93.0 \pm 0.5}$	83.5 ± 1.4	$\mathbf{91.8 \pm 1.1}$
ACMP-GAT (ours)	$\mathbf{82.3 \pm 0.5}$	$\mathbf{75.5 \pm 1.0}$	$\mathbf{79.4 \pm 0.4}$	91.8 ± 0.1	$\mathbf{84.4 \pm 1.6}$	91.1 ± 0.7

Table 2: Test accuracy and std for 10 initialization and 100 random train-val-test splits on six benchmark graph node classification tasks. We show the best three methods in red (First), blue (Second), and violet (Third).

Heterophilic datasets We evaluate ACMP-GCN on the heterophilic graphs; Cornell, Texas and Wisconsin from the WebKB dataset[3]. In this case, the assumption of common neighbors does not hold. The poor performance of GCN and GAT models shown in Table 1 indicates that many GNN models struggle in this setting. Introducing the Allen-Cahn term can improve the performance of GNNs on heteroplilic datasets significantly. ACMP-GCN scores 30% higher than the original GCN for the Texas dataset which has the smallest homophily level among the datasets in the table.

Attractive and Repulsive interpretation As shown in Table 1 and Table 2, ACMP-GCN and ACMP-GAT achieve better performance than GCN and GAT on both homophilic and heterophilic datasets. The majority of $a_{i,j} - \beta$ in the homophilic are positive, which means most nodes are attracted to each other. Conversely, most $a_{i,j} - \beta$ for the heterophilic are negative, which means that all the nodes are repelling their neighbors. Several GNNs exploiting multi-hop information can achieve high performance in node classification [72, 43]. However, high-order neighbor information will make the adjacency matrix dense and therefore can not be extended to large graphs. In our model, we take only one-hop information into account and add repulsive force ($\beta > 0$) to message passing, which has achieved the same or higher level of accuracy as multi-hop models in heterophilic dataset.

7 Related work

Neural Differential equations [38] provided an intimate survey on neural differential equations. The topic of neural ODEs becomes an emerging field since E [27] and Chen et al.'s work [19], with

[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

many follow-up works. Augmented [26] and regularized [30] forms are explored to enhance network expression power. Neural ODEs have been introduced into the GNN field: [7] used continuous residual modules for graph kernels; [58] extended the framework of GNN to continuous time. [62] applied Hamiltonian mechanics to graph networks to predict future states. For neural PDEs [60] attempted to use deep learning to solve PDEs. [13] used message passing to solve PDEs numerically. [10] developed a hybrid (graph) neural network embedded with PDE. GRAND [17] approached graph deep learning as a continuous diffusion process and propagated GNN by graph diffusion equation. [28] combined diffusion and wave PDEs for GNNs, and GCON [40] generalized this method. The latter employed a second-order system to conquer oversmoothing.

Flocking and Consensus In the literature the microscopic (agent-based particle systems) modeling of flocking and consensus has been extensively studied [22, 23, 21, 50, 16] with asymptotic estimates [50, 29, 34, 35, 37]. The flocking problem is to some degree similar to the general consensus problem [55] which studies the emergent behaviours for multi-agent systems. The Cucker-Smale model [22] is a famous model in this field considering a second-order system adopting to classical dynamics.

8 Conclusion

We develop a new message passing method based on Allen-Cahn particle system evolution. The new scheme treats the graph learning as particle evolution, and exploits the flexible design of particle equations. The proposed ACMP inherits the merits of separability and boundedness of the particle equation. The separability comes from introducing the attractive/repulsive force, with an Allen-Cahn term of double-well potential that serves to bound the node feature in the evolution and keeps the clusters near the local equilibria of the well thus prevents occurrence of oversmoothing. Experiments show excellent performance of the model for real datasets with various homophilic difficulty.

References

[1] Chapter 7 - phase equilibria of pure materials. In Bruce Fegley, editor, *Practical Chemical Thermodynamics for Geoscientists*, pages 225–286. Academic Press, Boston, 2013.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[3] Giacomo Albi, Nicola Bellomo, Luisa Fermo, S-Y Ha, J Kim, Lorenzo Pareschi, David Poyato, and Juan Soler. Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives. *Mathematical Models and Methods in Applied Sciences*, 29(10):1901–2005, 2019.

[4] Samuel M Allen and John W Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

[5] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *ICLR*, 2021.

[6] Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, pages 1–10, 2021.

[7] Pedro HC Avelar, Anderson R Tavares, Marco Gori, and Luis C Lamb. Discrete and continuous deep residual learning over graphs. *arXiv preprint arXiv:1911.09554*, 2019.

[8] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a 3-track network. *Science*, 373:871–876, 2021.

[9] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas

Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[10] Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In *ICML*, pages 2402–2411, 2020.

[11] Nicola Bellomo and Seung-Yeal Ha. A quest toward a mathematical theory of the dynamics of swarms. *Mathematical Models and Methods in Applied Sciences*, 27(04):745–770, 2017.

[12] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. In *ICLR*, 2022.

[13] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *ICLR*, 2022.

[14] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv preprint arXiv:2104.13478*, 2021.

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

[16] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591, 2009.

[17] Benjamin Paul Chamberlain, James Rowbottom, Maria I. Gorinova, Stefan D Webb, Emanuele Rossi, and Michael M. Bronstein. GRAND: Graph neural diffusion. In *ICML*, 2021.

[18] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, pages 1725–1735. PMLR, 2020.

[19] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, volume 31, 2018.

[20] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

[21] Felipe Cucker and Ernesto Mordecki. Flocking in noisy environments. *Journal de Mathématiques Pures et Appliquées*, 89(3):278–296, 2008.

[22] Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.

[23] Felipe Cucker and Steve Smale. On the mathematics of emergence. *Japanese Journal of Mathematics*, 2(1):197–227, 2007.

[24] Pierre Degond and Sébastien Motsch. Large scale dynamics of the persistent turning walker model of fish behavior. *Journal of Statistical Physics*, 131(6):989–1021, 2008.

[25] Renjun Duan, Massimo Fornasier, and Giuseppe Toscani. A kinetic flocking model with diffusion. *Communications in Mathematical Physics*, 300(1):95–145, 2010.

[26] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *NeurIPS*, volume 32, 2019.

[27] Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11, 2017.

[28] Moshe Eliasof, Eldad Haber, and Eran Treister. PDE-GCN: Novel architectures for graph neural networks motivated by partial differential equations. In *NeurIPS*, 2021.

[29] Di Fang, Seung-Yeal Ha, and Shi Jin. Emergent behaviors of the cucker–smale ensemble under attractive–repulsive couplings and rayleigh frictions. *Mathematical Models and Methods in Applied Sciences*, 29(07):1349–1385, 2019.

[30] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ODE: the world of Jacobian and kinetic regularization. In *ICML*, pages 3154–3164, 2020.

[31] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[32] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *NeurIPS*, 2019.

[33] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[34] Seung-Yeal Ha, Taeyoung Ha, and Jong-Ho Kim. Asymptotic dynamics for the Cucker-Smale-type model with the Rayleigh friction. *Journal of Physics A: Mathematical and Theoretical*, 43(31):315201, 2010.

[35] Seung-Yeal Ha and Eitan Tadmor. From particle to kinetic and hydrodynamic descriptions of flocking. *Kinetic & Related Models*, 1(3):415, 2008.

[36] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[37] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[38] Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

[39] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[40] T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-Coupled Oscillator Networks. In *ICML*, 2022.

[41] Naomi Ehrich Leonard, Derek A Paley, Francois Lekien, Rodolphe Sepulchre, David M Fratantoni, and Russ E Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[42] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *ICML*, pages 3276–3285, 2018.

[43] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

[44] Ettore Majorana. Il valore delle leggi statistiche nella fisica e nelle scienze sociali, Scientia, Quarta serie, Febbraio-Marzo 1942 pp. 58. English translation in Ettore Majorana, the value of statistical laws in physics and social sciences. *Quantitative Finance*, 5:133, 2005.

[45] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[46] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5115–5124, 2017.

[47] Jacob L Moreno. *Sociometry, experimental method and the science of society*. Lulu. com, 1951.

[48] Jacob Levy Moreno. Who shall survive?: A new approach to the problem of human interrelations. 1934.

[49] Sebastien Motsch and Eitan Tadmor. A new model for self-organized dynamics and its flocking behavior. *Journal of Statistical Physics*, 144(5):923–947, 2011.

[50] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM Review*, 56(4):577–621, 2014.

[51] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, page 1, 2012.

[52] Mark Ed Newman, Albert-László Ed Barabási, and Duncan J Watts. *The structure and dynamics of networks.* Princeton university press, 2006.

[53] Alexander Norcliffe, Cristian Bodnar, Ben Day, Nikola Simidjievski, and Pietro Liò. On second order behaviour in augmented neural ODEs. In *NeurIPS*, volume 33, pages 5911–5921, 2020.

[54] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

[55] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[56] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2019.

[57] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: geometric graph convolutional networks. In *ICLR*, 2020.

[58] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *AAAI Workshop on Deep Learning on Graphs: Methodologies and Applications*, 2020.

[59] Anton V Proskurnikov and Roberto Tempo. A tutorial on modeling and analysis of dynamic social networks. Part I. *Annual Reviews in Control*, 43:65–79, 2017.

[60] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[61] Loan Rayleigh. *JWS, The Theory of Sound, vol. 1*. Macmillan, London (reprinted Dover, New York, 1945), 1894.

[62] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ODE integrators. *NeurIPS Workshop on Machine Learning and the Physical Sciences*, 2019.

[63] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

[64] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[65] John Toner and Yuhai Tu. Flocks, herds, and schools: A quantitative theory of flocking. *Physical Review E*, 58(4):4828, 1998.

[66] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[67] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.

[68] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[69] Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *ICML*, pages 10432–10441, 2020.

[70] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018.

[71] Lingxiao Zhao and Leman Akoglu. PairNorm: Tackling oversmoothing in GNNs. In *ICLR*, 2020.

[72] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, volume 33, pages 7793–7804, 2020.

A The Gradient Flow Interpretation of Allen-Cahn

The Allen–Cahn equation (after John W. Cahn and Sam Allen [4]) is a phase transition model. It describes a reaction-diffusion process including order-disorder transitions. Let $\Omega \subseteq \mathbb{R}^N$ of a "binary mixture": Ice in solid phase $(+1)$ and water in liquid phase (-1). The configuration of this mixture in Ω can be described as a function

$$u^*(x) = \begin{cases} +1 & \textit{in } \Lambda \\ -1 & \textit{otherwise}, \end{cases} \tag{11}$$

where Λ is some open subset of Ω. We will say that u^* is the phase function.

Set the double-well potential $W(u) = \frac{1}{4}(1-u^2)^2$. Define the energy potential $\Phi : L^2(\Omega) \to \mathbb{R}_+$

$$\Phi(\mu, u) = \int (\mu^2 |\nabla u|^2 + W(u)). \tag{12}$$

Then, we can calculate the variation $\delta\Phi$

$$-\frac{\delta \Phi}{\delta u} = -\mu^2 \Delta u + W'(u). \tag{13}$$

To find a dynamic so that u can achieve some stable state u^*, one can choose the gradient direction of the potential energy as the search direction. Note that $\nabla \Phi = \frac{\delta \Phi}{\delta u}$ in $L_2(\Omega)$, then one can design such gradient flow in $L^2(\Omega)$:

$$u_t = -\nabla \Phi = -\frac{\delta \Phi}{\delta u} = \mu^2 \Delta u - W'(u), \tag{14}$$

which is the Allen-Cahn equation:

$$u_t = \mu^2 \Delta u + u(1-u)(1+u). \tag{15}$$

B Proofs of Propositions in Section 4

We assume that $a_{i,j}$ is symmetric, and $a_{i,j} > 0$ if $a_{i,j} \neq 0$. This condition means that graph is undirected. Since we deal with each channel independently, we abuse the notation to let x_i denote one feature component of node $\mathbf{x}_i$ to simplifying the notation in proofs.

B.1 The GRAND model

First, we consider the *oversmoothing* phenomenon if there is only the diffusion process with diffusion coefficients independent of x_i, which is a specific model of graph diffusion network (GRAND) [17],

$$\dot{x}_i = \alpha \sum_{j:(i,j)\in\mathcal{E}} a_{i,j}(x_j - x_i). \tag{16}$$

Proposition 1 Let $\mathbf{D}$ denote the degree matrix, i.e., $\mathbf{D} := \mathrm{diag}(d_1, \cdots, d_N)$, where $d_i = \sum_j a_{i,j}$. Then $\mathbf{D} - \mathbf{A}$ is symmetric positive semi-definite with the eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{\max} < \infty$. Let $\lambda_{\min} > 0$ be the smallest positive eigenvalue, then for all $t \geq 0$, there exists a constant $C > 0$ such that $\mathbf{E}(\mathbf{x}(t)) \leq C \exp(-\lambda_{\min}^2 t)$.

Proof. Let $\mathcal{L} := \mathbf{D} - \mathbf{A}$, we have,

$$\mathbf{x}(t) = \mathbf{x}(0) e^{-\mathcal{L}}.$$

Using eigenvalue decomposition, the solution $x(t)$ writes

$$\mathbf{x}(t) = \mathbf{U}^\top e^{-\mathbf{\Lambda} t} \mathbf{U} \mathbf{x}(0) \tag{17}$$

Since the Dirichlet energy can also be written as

$$\mathbf{E}(\mathbf{x}(t)) = \mathbf{x}(t)^\top \mathcal{L} \mathbf{x}(t), \tag{18}$$

Taking (17) to (18) gives

$$\mathbf{E}(\mathbf{x}(t)) = \mathbf{x}(0)^\top \mathbf{U}^\top e^{-\mathbf{\Lambda} t} \mathbf{\Lambda} e^{-\mathbf{\Lambda} t} \mathbf{U} \mathbf{x}(0). \tag{19}$$

Therefore, $\mathbf{E}(\mathbf{x}(t)) \leq C \exp(-\lambda_{\min}^2 t)$ for some constant $C > 0$. ■

Propsition 7 *We also consider a more general case,*

$$\frac{d}{dt}x_i(t) = \sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i), \tag{20}$$

with $a(x_i, x_j) = a(x_j, x_i) \geq a_{\min} > 0$, *for any* $x_i,\ x_j$.

Let the mass center $x_c = \frac{1}{N}\sum_{i\in\mathcal{V}} x_i$. *From the symmetry of* $a(x_i, x_j)$ *and (20), we obtain* $\mathrm{d}x_c/\mathrm{d}t = 0$ *for any* $t > 0$. *Without loss of generality, we may assume*

$$x_c(0) = 0, \tag{21}$$

and graph $\mathcal{G}$ *is connected, i.e.,* $\forall (i,j) \in \mathcal{V}\times\mathcal{V}$, $\mathcal{G}$ *contains a path from* i *to* j. *Then we have,* $\|x(t)\|^2 \leq \|x(0)\|^2 e^{-2a_{\min}\lambda_{\min}t}$ *and* $\mathbf{E}(x(t)) \leq \lambda_{\max}\|x(0)\|^2 e^{-2a_{\min}\lambda_{\min}t}$. *Note that the above estimates hold true for any initial condition* $x_c(0) = c$, *since* x *satisfies the ODE system (20) up to a constant. If* $x_c(0) = c$, x_i *will converge to* c *in time. If* $\mathcal{G}$ *is not connected, then we just need to consider each connected sub-graph separately with the assumption* $x_{c'}(0) = \frac{1}{N'}\sum_{i\in\mathcal{V}'} x_i = c'$ *for each sub-graph* $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. x_i' *in each sub-graph will converge to constant* c' *independently.*

Proof. *We multiply* x_i *on both sides of the equation (20) and sum over* x_i *to obtain*

$$x_i\frac{dx_i}{dt} = \sum_{j\in N(j)} a\left(x_i, x_j\right)\left(x_j - x_i\right)x_i \tag{22}$$

$$\Rightarrow \quad \frac{d}{dt}\|x\|^2 = -2\sum_{(i,j)\in\mathcal{E}} a\left(x_i, x_j\right)\left(x_j - x_i\right)^2 \tag{23}$$

$$\Rightarrow \quad \frac{d}{dt}\|x\|^2 \leqslant -2a_{\min}\sum_{(i,j)\in\mathcal{E}} \left(x_j - x_i\right)^2 \tag{24}$$

The RHS in (24) can be written in matrix form with $\mathcal{L} := \mathbf{D} - \mathbf{A}$,

$$\sum_{(i,j)\in\mathcal{E}} \left(x_j - x_i\right)^2 = \sum_{(i,j)\in\mathcal{V}\times\mathcal{V}} a_{i,j}\left(x_j - x_i\right)^2 = x^\top\mathcal{L}x.$$

Since $\mathcal{G}$ *is a connected graph,* $\mathbf{1}$ *is the only eigenvector consisting of the kernel space of* $\mathcal{L}$, *therefore,* $x^T\mathcal{L}x \geq \lambda_{\min}\|x\|^2$ *for any* x *satisfying* $\sum_{i\in\mathcal{V}} x_i = 0$. *Then, (24) leads to*

$$\frac{d}{dt}\|x\|^2 \leqslant -2a_{\min}\lambda_{\min}\|x\|^2. \tag{25}$$

This yields the decay estimates for $\|x\|$ *and* $\mathbf{E}(x(t))$*:*

$$\|x(t)\|^2 \leq \|x(0)\|^2 e^{-2a_{\min}\lambda_{\min}t}, \quad \mathbf{E}(x(t)) \leq \lambda_{\max}\|x(0)\|^2 e^{-2a_{\min}\lambda_{\min}t}.$$

■

B.2 The model with the Allen-Cahn term

Next, we consider the case $\beta = 0$ but with the Allen-Cahn term:

$$\begin{cases} \dfrac{d}{dt}x_i(t) = \alpha\displaystyle\sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i) + \delta x_i\left(1 - x_i^2\right), \\ a(x_i, x_j) = a(x_j, x_i) \geq 0, \quad \forall i, j \in \mathcal{V} \quad \displaystyle\sum_i a(x_i, x_j) = 1,\ \forall j \in \mathcal{V}. \end{cases} \tag{26}$$

For (26), we can assert that the stable equilibrium of any x_i is limited in the interval $[-1, 1]$.

Propsition 8 *Suppose* $\mathbf{x}^* = (x_1^*, \ldots, x_N^*)$ *is a global equilibrium (or steady state solution) of (26) on* $\mathbb{R}$ *and* $\mathbf{x}$, *then* $x_i^* \in [-1, 1]$.

Proof. Suppose x^* achieves the equilibrium of (26), and $x_k^* \geq x_i^* \ \forall i$. If $x_k^* > 1$, then $\alpha \sum_{j:(k,j)\in\mathcal{E}} a(x_k^*, x_j^*)(x_j^* - x_k^*) \leq 0$ and $x_k^*(1 - x_k^{*2}) < 0$, which contradicts with $\frac{\partial}{\partial t} x_k^* = 0$. ■

The emergence of clusters depends on the distribution of initial features. If all the initial features are in only one potential well, then intuitively it is impossible to produce more than one cluster in the dynamics (26). As a simple transference of Lemma 3.2 in [34], we can prove this. Set

$$x^M(t) := \max_i x_i(t), \quad x^m(t) := \min_i x_i(t), \tag{27}$$

where x_i is still some component of node feature $\mathbf{x}_i$. Assume x^m, x^M are both Lipschitz continuous and therefore they are almost differentiable everywhere in time t.

Propsition 9 *Let $\{x_i\}$ be the solutions of (26), then the following holds.*
(i) If $x^m(0) > 0$, then $x^m(t) \geq 0$ for all $t > 0$.
(ii) If $x^M(0) < 0$, then $x^M(t) \leq 0$ for all $t > 0$.

Proof. The proof was essentially given by [34]. For the sake of completeness, we give a proof here. (i) If $x^m(0) > 0$, we assert there exists a time sequence $\{t_j\}_{j=0}^{\infty}$ satisfying $t_0 = 0 < t_1 < \cdots < t_j < \ldots$, $x^m(t)$ is differentiable in each time interval (t_{j-1}, t_j) and $x_i^m \geq 0$ when $t \in [0, t_1]$. By induction, firstly we set

$$x^m(t) \geq 0, \quad t \in [0, t_l].$$

If x^m becomes negative in the time interval (t_l, t_{l+1}) there exists $t^* \in (t_l, t_{l+1})$ such that $x^m(t^*) = 0$ by the continuity of $x^m(t)$. One can assume $x_m(t) \equiv x_i(t)$ for some node x_i in some time interval subset to (t_l, t_{l+1}). At that moment,

$$\begin{aligned} \frac{dx_i}{dt}(t^*) &= \alpha \sum_j a(x_j, x_i)(x_j(t^*) - x_i(t^*)) + \delta x_i(t^*)(1 - x_i^2(t^*)) \\ &= \alpha \sum_j a(x_j, x_i) x_j(t^*) \\ &\geq 0. \end{aligned} \tag{28}$$

Hence the trajectory x^m becomes non-decreasing at $t = t^*$. By induction, we derive (i).

(ii) can be proved by the same argument as those for (i). ■

Now we consider the second kinetic model (10). We can prove that if any particle x_i gets caught in one potential well, then it will not escape from that well.

Proof of Proposition 6. For the $\beta = 0$ case, assume $x_i = -1 + \epsilon$ for $\epsilon \leq \delta' < 1$ at a certain time t_0, that is, $x_i \in [-1, -1 + \delta')$. We want to show $\frac{dx_i}{dt}\big|_{t=t_0} < 0$, which means

$$\alpha \sum_{j\in\mathcal{N}_i} a_{i,j}(x_j - x_i)(1 - x_i^2)^2 < -\delta x_i(1 - x_i^2).$$

By $\sum_{j\in\mathcal{N}_i} a_{i,j} = 1$ from (26), the above inequality is equivalent to

$$\sum_{j\in\mathcal{N}_i} a_{i,j} x_j < \frac{\delta}{\alpha} \frac{1-\epsilon}{2-\epsilon} \frac{1}{\epsilon} + \epsilon - 1 \leq \frac{\delta}{2\alpha} \frac{1}{\epsilon} + \epsilon - 1 \leq \frac{\delta}{2\alpha\epsilon}. \tag{29}$$

Since $\{x_j\}_{j=1}^N$ are bounded (See Proposition 2.), (29) is satisfied for a sufficiently small δ'. The other case $x_i = 1 - \epsilon$ can be similarly proved.
For the $\beta \neq 0$ case, we also assume $x_i = -1 + \epsilon$ for $\epsilon \leq \delta' < 1$ at a certain time t_0. Similarly with (29), we have

$$\sum_{j\in\mathcal{N}_i} (a_{i,j} - \beta) x_j < \frac{\delta}{\alpha} \frac{1-\epsilon}{2-\epsilon} \frac{1}{\epsilon} + (1 - d_i\beta)(\epsilon - 1) \leq \frac{\delta}{2\alpha\epsilon} + d_i\beta - 1 + \epsilon(1 - d_i\beta).$$

By the boundedness of $\{x_j\}_{j=1}^N$, a properly small δ' can be found. ■

B.3 The Attractive-repulsive Model

We first show that the solution features of graph in the Allen-Cahn model below is bounded. For simplicity of the proof, we rewrite (8) in component form where we let $a(x_i, x_j) := a_{i,j} - \beta_{i,j}$:

$$\frac{d}{dt}x_i(t) = \alpha \sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i) + \delta x_i \left(1 - x_i^2\right). \tag{30}$$

Model (30) allows negative $a(x_i, x_j)$ which is different from the condition in (26).

Proof of Proposition 2 and Proposition 3. We multiply x_i on both sides of the following equation and sum over x_i to obtain

$$\begin{aligned}
&\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i) - x_i^3 + x_i \\
\Rightarrow &\frac{1}{2}\frac{\mathrm{d}x_i^2}{\mathrm{d}t} = \sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)\,x_i - x_i^4 + x_i^2 \\
\Rightarrow &\frac{1}{2}\sum_{i\in\mathcal{V}}\frac{\mathrm{d}x_i^2}{\mathrm{d}t} = -\sum_{i\in\mathcal{V}}\left(\sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)\,x_i - x_i^4 + x_i^2\right).
\end{aligned} \tag{31}$$

By grouping $a(x_i, x_j)\,(x_j - x_i)\,x_i$, then

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|x\|^2 = -\frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)^2 - \sum_{i\in\mathcal{V}} x_i^4 + \|x\|^2. \tag{32}$$

Note that $a(x_i, x_j)$ are bounded for any (x_i, x_j). Let the $|a(x_i, x_j)| < D_1$ for a constant D_1 depending on hyper-parameters $\beta_{i,j}$. By the Cauchy-Schwarz inequality,

$$|a(x_i, x_j)(x_j - x_i)^2| \le 2D_1(x_i^2 + x_j^2).$$

Hence

$$-\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)^2 \le c_4\|x\|^2.$$

Also, $\sum_{i\in\mathcal{V}} x_i^4 \ge c_3\|x\|^4$ for a constant c_3 depending only on N. Taking the above estimates to (32) gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\|x\|^2 \le -2c_3\|x\|^4 + (c_4 + 2)\|x\|^2.$$

If $\|x\|$ blows up for $t > 0$, the $\|x\| \to \infty$ as time t increases, and $\frac{d}{dt}\|x\|^2 > 0$ for all t before the blowing-up time T_{end}. However, one can find a $t^* < T_{\text{end}}$ such that $\|x(t^*)\|$ is large enough and

$$-2c_3\|x(t^*)\|^4 + (c_4 + 2)\|x(t^*)\|^2 < 0,$$

which produces a contradiction. Thus, $\|x\| \le c_5$ for a constant c_5 only depending on N and D_1 and

$$\mathbf{E}(x) \le \lambda_{\max}\|x\|^2 \le \lambda_{\max}c_5,$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathcal{L} := \mathbf{D} - \mathbf{A}$. Thus, we proved the assertion in Proposition 3.

■

For an attractive-repulsive system, we change β into $\beta_{i,j}$. We here show the long-time behaviour of the Cucker-Smale model [22] for strength coupling (α, δ) that satisfies the following condition: there exists $\{\beta_{i,j}\}$ such that $\mathcal{I} := \{1, \dots, N\}$ can be divided into two disjoint groups $\mathcal{I}_1, \mathcal{I}_2$:

$$\begin{aligned}
&0 < S \le \overline{a}_{i,j} \ \text{ with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \text{for } i, j \in I_1, \\
&0 < S \le \overline{a}_{i,j} \ \text{ with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \text{for } i, j \in I_2, \\
&0 \le \overline{a}_{i,j} \le D \ \text{ with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \textit{otherwise}, \\
&0 \le \overline{a}_{i,j} \le D \ \text{ with } -\overline{a}_{i,j} := a_{i,j} - \beta_{i,j} \quad \textit{otherwise},
\end{aligned} \tag{33}$$

where S, D are independent of time t. This system can be proved to produce bi-cluster stable equilibria. In ACMP-GCN, $\overline{a}_{i,j}^{\text{GCN}} := a_{i,j}^{\text{GCN}} - \beta_{i,j}$ are parameters and we omit the superscript GCN in $\overline{a}_{i,j}(t)$ in the following.

Remark 1 *For ACMP-GAT, $(a_{i,j}^{\mathrm{attn}} - \beta_{i,j})$ may depend on time t. In this case, we need to assume $\overline{a}_{i,j}(t)$ always satisfies (33) for all t. The following proof still works.*

Denote node features indexed by $\mathcal{I}_1, \mathcal{I}_2$ by $x_i^{(1)}, x_i^{(2)}$ respectively. Then (30) can be rewritten as

$$\begin{cases} \dfrac{d}{dt}x_i^{(1)} = \alpha \displaystyle\sum_k^{N_1} \overline{a}_{k,i}(x_k^{(1)} - x_i^{(1)}) - \alpha \sum_k^{N_2} \overline{a}_{k,i}(x_k^{(2)} - x_i^{(1)}) + \delta x_i^{(1)}(1 - (x_i^{(1)})^2), \ i = 1, \dots, N_1 \\ \dfrac{d}{dt}x_j^{(2)} = \alpha \displaystyle\sum_k^{N_2} \overline{a}_{k,j}(x_k^{(1)} - x_i^{(1)}) - \alpha \sum_k^{N_1} \overline{a}_{k,j}(x_k^{(2)} - x_j^{(1)}) + \delta x_j^{(2)}(1 - (x_j^{(2)})^2), \ j = 1, \dots, N_2. \end{cases} \tag{34}$$

For the attractive-repulsive model (34), we can refer to the results in [29] and obtain the following proposition by following the proof of its Theorem 5.1.

Definition 2 *Let $\{x_i\}_{i=1}^N = \{x_i^{(1)} \cup x_j^{(2)}\}$ be a solution to system (30). Then, the solution tends to the bi-cluster flocking if*

$$\begin{aligned} &(i) \sup_{0\le t<\infty} \max_{1\le i,j\in N_1} |x_i^{(1)}(t) - x_j^{(1)}(t)| < \infty, \quad \sup_{0\le t<\infty} \max_{1\le i,j\in N_2} |x_i^{(2)}(t) - x_j^{(2)}(t)| < \infty; \\ &(ii) \ \exists\, C, T^{**} > 0 \min_{i\in\mathcal{I}_1, j\in\mathcal{I}_2} \{|x_i^{(1)}(t) - x_j^{(2)}(t)|\} \ge C, \quad \forall t > T^{**}. \end{aligned} \tag{35}$$

We define the following notations for further proof:

$$\begin{aligned} &V := \{\textit{nodes indexed by } \mathcal{I}_1\}, \quad W := \{\textit{nodes indexed by } \mathcal{I}_2\}, \\ &N_1 := |V|, \quad N_2 := |W|, \\ &\widehat{x^{(1)}} := x_i^{(1)} - x_c^{(1)}, \quad \widehat{x^{(2)}} := x_i^{(2)} - x_c^{(2)}, \\ &x_c^{(1)} := \frac{1}{N_1}\sum_{i=1}^{N_1} x_i^{(1)}, \quad x_c^{(2)} := \frac{1}{N_2}\sum_{i=1}^{N_2} x_i^{(2)}, \\ &M_2(V) := \frac{1}{N_1}\sum_{i=1}^{N_1}(x_i^{(1)})^2, \quad M_2(W) := \frac{1}{N_2}\sum_{i=1}^{N_2}(x_i^{(2)})^2, \\ &M_2 := M_2(V) + M_2(W), \\ &\widehat{M_2} := M_2(\widehat{V}) + M_2(\widehat{W}). \end{aligned}$$

Propsition 10 *Let $\overline{a}_{i,j}$ be given by (33). Suppose that $\alpha, \delta > 0$, the initial centers of two groups are separated enough and the intra-interaction outweighs the inter-interaction, namely*

$$\|x_c^{(1)}(0) - x_c^{(2)}(0)\| \gg 1, \quad (\delta + \alpha D \max\{N_1, N_2\} - \alpha S \min\{N_1, N_2\}) \le -\eta, \tag{36}$$

Then, the system has a bi-cluster flocking.

Remark 2 *(36) indicates that the repulsive force between the particles should be stronger than the attractive ($S > D$).*

To prove Proposition 10, we need the following two lemmas, which we would postpone to prove.

Lemma 1 *Let $\{x_i\}$ be a solution to (34). Then $\widehat{M}_2$ satisfies*

$$\begin{aligned}\frac{d}{dt}M_2 =& -\frac{\alpha}{N_1}\sum_{i,k}^{N_1}\overline{a}_{k,i}(x_k^{(1)}-x_i^{(1)})^2-\frac{2\alpha}{N_1}\sum_{k}^{N_2}\sum_{i}^{N_1}\overline{a}_{i,k}(x_k^{(2)}-x_i^{(1)})x_i^{(1)}\\&+\frac{2\delta}{N_1}\sum_{i}^{N_1}(x_i^{(1)})^2(1-(x_i^{(1)})^2)\\&-\frac{\alpha}{N_2}\sum_{j,k}^{N_2}\overline{a}_{k,j}(x_k^{(2)}-x_j^{(2)})^2-\frac{2\alpha}{N_2}\sum_{j}^{N_2}\sum_{i}^{N_2}\overline{a}_{j,k}(x_k^{(1)}-x_j^{(2)})x_j^{(2)}\\&+\frac{2\delta}{N_2}\sum_{i}^{N_2}(x_j^{(2)})^2(1-(x_j^{(2)})^2).\end{aligned}\tag{37}$$

Suppose that the system parameters satisfy

$$S\geq 0,\quad D>0,\quad \delta>0,$$

then there exists a positive constant M_2^∞ such that

$$\sup_{0\leq t<\infty}M_2(t)\leq M_2^\infty<\infty.\tag{38}$$

Lemma 2 *Let $\{x_i\}$ be a solution to (34) with $\delta>0$. Then $\widehat{M}_2$ satisfies*

$$\frac{d}{dt}\widehat{M}_2\leq -2\eta\widehat{M}_2+2\alpha D\zeta|x_c^{(1)}-x_c^{(2)}|\sqrt{\widehat{M}_2},\tag{39}$$

where $\zeta=\max\{N_1,N_2\}$.

Proof of Proposition 10.
(a) (Uniform upper bound of $|x_c^{(1)}-x_c^{(2)}|$) By Cauchy's inequality and Lemma 1,

$$\begin{aligned}|x_c^{(1)}-x_c^{(2)}|&=\left|\frac{1}{N_1}\sum_{i=1}^{N_1}x_i^{(1)}-\frac{1}{N_2}\sum_{i=1}^{N_2}x_i^{(2)}\right|\\&\leq\frac{1}{N_1}\sum_{i=1}^{N_1}|x_i^{(1)}|+\frac{1}{N_2}\sum_{i=1}^{N_2}|x_i^{(2)}|\\&\leq 2\sqrt{\frac{1}{N_1}\sum_{i=1}^{N_1}(x_i^{(1)})^2+\frac{1}{N_2}\sum_{i=1}^{N_2}(x_i^{(2)})^2}\\&=2\sqrt{M_2(t)}\leq 2\sqrt{M_2^\infty}.\end{aligned}\tag{40}$$

(b) (Uniform boundedness of $\widehat{M}_2$) By Lemma 2 and (40),

$$\begin{aligned}\frac{d}{dt}\sqrt{\widehat{M}_2}&\leq-\eta\sqrt{\widehat{M}_2}+\alpha D\zeta|x_c^{(1)}-x_c^{(2)}|\\&\leq-\eta\sqrt{\widehat{M}_2}+2D\alpha\zeta\sqrt{M_2^\infty}.\end{aligned}\tag{41}$$

Use Gronwall's lemma to obtain

$$\begin{aligned}\sqrt{\widehat{M}_2(t)}&\leq\sqrt{\widehat{M}_2(0)}e^{-\eta t}+\frac{2D\alpha\zeta\sqrt{M_2^\infty}}{\eta}(1-e^{-\eta t})\\&\leq\max\left\{\sqrt{\widehat{M}_2(0)},\frac{2D\alpha\zeta\sqrt{M_2^\infty}}{\eta}\right\}:=C_3.\end{aligned}\tag{42}$$

(c) (Separation of the particles) By a similar estimate with Lemma 2, we have

$$\begin{aligned}\frac{d}{dt}|x_c^{(1)}-x_c^{(2)}|^2 =&2(x_c^{(1)}-x_c^{(2)})\frac{\alpha}{N_1}\sum_k^{N_2}\sum_i^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(2)}}+x_c^{(2)}-\widehat{x_i^{(1)}}-x_c^{(1)})\\ &-2(x_c^{(1)}-x_c^{(2)})\frac{\alpha}{N_2}\sum_k^{N_2}\sum_i^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(2)}}+x_c^{(2)}-\widehat{x_i^{(1)}}-x_c^{(1)})\\ &-\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^3(x_c^{(1)}-x_c^{(2)})+\frac{2\delta}{N_2}\sum_i^{N_2}(x_i^{(2)})^3(x_c^{(1)}-x_c^{(2)})\\ \geq&2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)\sum_i^{N_1}\sum_j^{N_2}\overline{a}_{i,j}(x_c^{(1)}-x_c^{(2)})^2+\mathcal{I}_{c1}+\mathcal{I}_{c2}.\end{aligned}\tag{43}$$

By the Cauchy-Schwarz inequality,

$$\begin{aligned}|\mathcal{I}_{c1}| &= -2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)\sum_i^{N_1}\sum_j^{N_2}\overline{a}_{i,j}(\widehat{x_j^{(2)}}-\widehat{x_i^{(1)}})(x_c^{(1)}-x_c^{(2)})\\ &\leq 2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)D\sqrt{N_1N_2}|x_c^{(1)}-x_c^{(2)}|\sqrt{\sum_{i,j}^{N_1,N_2}(\widehat{x_j^{(2)}}-\widehat{x_i^{(1)}})^2}\\ &= 2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)DN_1N_2|x_c^{(1)}-x_c^{(2)}|\widehat{M_2}.\end{aligned}\tag{44}$$

For $\mathcal{I}_{c2}:=\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^3(x_c^{(1)}-x_c^{(2)})+\frac{2\delta}{N_2}\sum_i^{N_2}(x_i^{(2)})^3(x_c^{(1)}-x_c^{(2)})$, note that

$$\begin{aligned}\left|\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^3\right|&\leq\delta|x_i^{(1)}|M_2(V)\leq\delta\sqrt{N_1}M_2(V)^{\frac{3}{2}},\\ \left|\frac{2\delta}{N_2}\sum_i^{N_2}(x_i^{(2)})^3\right|&\leq\delta|x_i^{(2)}|M_2(V)\leq\delta\sqrt{N_1}M_2(W)^{\frac{3}{2}}.\end{aligned}$$

Then, one gets

$$\begin{aligned}\mathcal{I}_{c2}&\geq-2\left|x_c^{(1)}-x_c^{(2)}\right|\left|\frac{\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^3+\frac{\delta}{N_2}\sum_i^{N_2}(x_i^{(2)})^3\right|\\ &\geq-2\left|x_c^{(1)}-x_c^{(2)}\right|\delta\sqrt{\max\{N_1,N_2\}}M_2(t)^{\frac{3}{2}}.\end{aligned}\tag{45}$$

Hence,

$$\begin{aligned}\frac{d}{dt}|x_c^{(1)}-x_c^{(2)}|^2\geq&\left[\left(2\alpha(\frac{1}{N_1}+\frac{1}{N_2})\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{i,j}\right)+\delta\right]|x_c^{(1)}-x_c^{(2)}|^2\\ &-\alpha D(N_1+N_2)\sqrt{\widehat{M_2}}-\delta\sqrt{\max\{N_1,N_2\}}M_2^{\frac{3}{2}}.\end{aligned}\tag{46}$$

Combining with Lemma 1 and (42), one obtains the estimate

$$\begin{aligned}\frac{d}{dt}|x_c^{(1)}-x_c^{(2)}|\geq&\left(2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)\sum_i^{N_1}\sum_j^{N_2}\overline{a}_{i,j}+\delta\right)|x_c^{(1)}-x_c^{(2)}|\\ &-\alpha D(N_1+N_2)C_3-\delta\sqrt{\max\{N_1,N_2\}}(M_2^\infty)^{\frac{3}{2}}.\end{aligned}\tag{47}$$

By Gronwall's lemma, if the initial data satisfy:

$$|x_c^{(1)}(0)-x_c^{(2)}(0)|\geq\frac{\alpha D(N_1+N_2)C_3+\delta\sqrt{\max\{N_1,N_2\}}(M_2^\infty)^{\frac{3}{2}}}{\delta}:=\frac{C_4}{\delta},\tag{48}$$

then,

$$|x_c^{(1)}(t) - x_c^{(2)}(t)| \geq \frac{C_4}{\delta} + (|v_c(0) - w_c(0)| - \frac{C_4}{\delta})e^{\delta t} \geq \frac{C_4}{\delta}. \tag{49}$$

(d) For any $i = 1, \dots N_1, j = 1, \dots, N_2$,

$$\begin{aligned}
|x_i^{(1)}(t) - x_j^{(1)}(t)| &\geq |x_c^{(1)}(t) - x_c^{(2)}(t)| - |\widehat{x^{(1)}}_i(t) - \widehat{x^{(2)}}_j(t)| \\
&\geq \frac{C_4}{\delta} - \sqrt{2\max\{N_1, N_2\}\widehat{M}_2} \\
&\geq \frac{C_4}{\delta} + \left(|x_c^{(1)}(0) - x_c^{(2)}(0)| - \frac{C_4}{\delta}\right) e^{\delta t} \\
&\quad - \sqrt{2\max\{N_1, N_2\}} \left(\sqrt{\widehat{M}_2(0)}e^{-\eta t} + \frac{\sqrt{M_2^\infty}}{\eta}(1 - e^{-\eta t})\right).
\end{aligned}$$

Then, there exists some time T^* such that $\forall t \geq T^*$,

$$|x_i^{(1)}(t) - x_j^{(2)}(t)| \geq C > 0, \quad \forall i, j. \tag{50}$$

Combing with Proposition 2, we finish the proof. ■

Remark 3 *The proof of Proposition 5 is included in part (d) of proof of Proposition 4.*

Now suppose $\eta_2 := \sum_{i \in \mathcal{I}_1, j \in \mathcal{I}_2} a_{i,j} > 0$ in some channel, then the Dirichlet energy in this channel has a lower bound:

$$\begin{aligned}
\mathbf{E}(x) &= \frac{1}{N}\sum_{i,j} a_{i,j}(x_i - x_j)^2 \\
&= \frac{1}{N}\left[\sum_{i,j \in \mathcal{I}_1} a_{i,j}(x_i^{(1)} - x_j^{(1)})^2 + \sum_{i,j \in \mathcal{I}_2} a_{i,j}(x_i^{(2)} - x_j^{(2)})^2 + \sum_{i \in \mathcal{I}_1, j \in \mathcal{I}_2} a_{i,j}(x_i^{(1)} - x_j^{(2)})^2\right] \\
&\geq \frac{1}{N}\sum_{i \in \mathcal{I}_1, j \in \mathcal{I}_2} a_{i,j}(x_i^{(1)} - x_j^{(2)})^2 \\
&\geq \frac{C^2\eta_2}{N}.
\end{aligned} \tag{51}$$

Proof of Lemma 1.

$$\begin{aligned}
\frac{d}{dt}M_2(V) &= \frac{2}{N_1}\sum_{i=1}^{N_1} x_i^{(1)}\dot{x_i^{(1)}} \\
&= -\frac{\alpha}{N_1}\sum_{i,k}^{N_1} \overline{a}_{k,i}(x_k^{(1)} - x_i^{(1)})^2 - \frac{2\alpha}{N_1}\sum_{k}^{N_2}\sum_{i}^{N_1} \overline{a}_{i,k}(x_k^{(2)} - x_i^{(1)})x_i^{(1)} \\
&\quad + \frac{2\delta}{N_1}\sum_{i}^{N_1}(x_i^{(1)})^2(1 - (x_i^{(1)})^2).
\end{aligned} \tag{52}$$

Similarly,

$$\begin{aligned}
\frac{d}{dt}M_2(W) &= \frac{2}{N_2}\sum_{i=1}^{N_2} x_i^{(2)}\dot{x_i^{(2)}} \\
&= -\frac{\alpha}{N_2}\sum_{j,k}^{N_2} \overline{a}_{k,j}(x_k^{(2)} - x_j^{(2)})^2 - \frac{2\alpha}{N_2}\sum_{j}^{N_2}\sum_{i}^{N_2} \overline{a}_{j,k}(x_k^{(1)} - x_j^{(2)})x_j^{(2)} \\
&\quad + \frac{2\delta}{N_2}\sum_{i}^{N_2}(x_j^{(2)})^2(1 - (x_j^{(2)})^2).
\end{aligned} \tag{53}$$

Sum the $M_2(V)$ and $M_2(W)$. Note that $\overline{a}_{ij} = \overline{a}_{ji}$. Then

$$\begin{aligned}\frac{d}{dt}M_2 \leq & \frac{D\alpha}{N_1}\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\left((x_k^{(2)}-x_i^{(1)})^2+(x_i^{(1)})^2\right)+\frac{D\alpha}{N_2}\sum_{k=1}^{N_1}\sum_{j=1}^{N_2}\left((x_k^{(1)}-x_j^{(2)})^2+(x_j^{(2)})^2\right)\\ &+\frac{2\delta}{N_1}\sum_{i=1}^{N_1}(x_i^{(1)})^2(1-(x_i^{(1)})^2)+\frac{2\delta}{N_2}\sum_{i=1}^{N_2}(x_i^{(2)})^2(1-(x_i^{(2)})^2).\end{aligned} \tag{54}$$

By the Cauchy-Schwarz inequality,

$$\begin{aligned}&\left(\sum_{i=1}^{N_1}(x_i^{(1)})^2\right)^2 \leq N_1\sum_{i=1}^{N_1}(x_i^{(1)})^4, \quad \left(\sum_{i=1}^{N_1}(x_i^{(1)})^2\right)^2 \leq N_2\sum_{i=1}^{N_2}(x_i^{(2)})^4,\\ &(x_i^{(1)}-x_j^{(2)})^2 \leq 2((x_i^{(1)})^2+(x_j^{(2)})^2).\end{aligned}$$

These relations and (54) yield a Riccati-type differential inequality:

$$\begin{aligned}\frac{d}{dt}M_2 &\leq 2D\alpha N_2 M_2(W)+3D\alpha N_2 M_2(V)+2D\alpha N_1 M_2(V)+3D\alpha N_2 M_2(W)+2\delta M_2-\delta(M_2)^2\\ &\leq (\alpha C_m+2\delta)M_2-\delta(M_2)^2.\end{aligned} \tag{55}$$

Let y be a solution of the following ODE:

$$y' = \alpha C_m y - \delta y^2. \tag{56}$$

Then, the solution $y(t)$ to equation 56 satisfies

$$M_2(t) \leq y(t) \leq \max\left\{\frac{\alpha C_m}{\delta}+2, M_2(0)\right\} =: M_2^\infty. \tag{57}$$

■

Proof of Lemma 2. By computation,

$$\begin{aligned}\dot{x_c^{(1)}} &= \frac{1}{N_1}\sum_{i=1}^{N_1}\dot{x_i^{(1)}}\\ &= \frac{\alpha}{N_1}\sum_{i,k=1}^{N_1}\overline{a}_{k,i}(x_k^{(1)}-x_i^{(1)})-\frac{\alpha}{N_1}\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\overline{a}_{k,i}(x_k^{(2)}-x_i^{(1)})+\frac{\delta}{N_1}\sum_{i=1}^{N_1}x_i^{(1)}(1-(x_i^{(1)})^2)\\ &= -\frac{\alpha}{N_1}\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\overline{a}_{k,i}(x_k^{(2)}-x_i^{(1)})+\frac{\delta}{N_1}\sum_{i=1}^{N_1}x_i^{(1)}(1-(x_i^{(1)})^2).\end{aligned}$$

Note that $\dot{\widehat{x_i^{(1)}}} = \dot{x_i^{(1)}} - \dot{x_c^{(1)}}$. Take the inner product $2\widehat{x_i^{(1)}}$ with the above equation and sum it over all $i = 1, \ldots, N_1$, combining with $\sum \widehat{x_i^{(1)}} = 0$. Then,

$$\begin{aligned}\frac{d}{dt}M_2(\widehat{V}) &= \frac{1}{N_1}\left[-\alpha\sum_{i,k=1}^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(1)}}-\widehat{x_i^{(1)}})^2-2\alpha\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\overline{a}_{k,i}(x_k^{(2)}-x_i^{(1)})\widehat{x_i^{(1)}}+2\delta\sum_{i=1}^{N_1}\widehat{x_i^{(1)}}x_i^{(1)}(1-(x_i^{(1)})^2)\right]\\ &= \frac{1}{N_1}\left[-\alpha\sum_{i,k=1}^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(1)}}-\widehat{x_i^{(1)}})^2-2\alpha\sum_{i=1}^{N_1}\sum_{k=1}^{N_2}\overline{a}_{k,i}(x_c^{(2)}-x_c^{(1)}+\widehat{x_k^{(2)}}-\widehat{x_i^{(1)}})\widehat{x_i^{(1)}}\right]\\ &\quad+\frac{1}{N_1}2\delta\sum_{i=1}^{N_1}\widehat{x_i^{(1)}}x_i^{(1)}(1-(x_i^{(1)})^2).\end{aligned}$$

Similarly,

$$\frac{d}{dt}M_2(\widehat{W}) = \frac{1}{N_2}\left[-\alpha\sum_{i,k=1}^{N_2}\overline{a}_{k,i}(\widehat{x_k^{(2)}} - \widehat{x_i^{(2)}})^2 - 2\alpha\sum_{k=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{k,j}(x_c^{(1)} - x_c^{(2)} + \widehat{x_k^{(1)}} - \widehat{x_j^{(2)}})\widehat{x_j^{(2)}}\right]$$
$$+ \frac{1}{N_2}2\delta\sum_{i=1}^{N_2}\widehat{x_i^{(2)}}x_i^{(2)}(1-(x_i^{(2)})^2).$$

Combine the two equations,

$$\frac{d}{dt}\widehat{M_2} = \sum_{i=1}^{6} I_i, \tag{58}$$

where

$$\begin{aligned}
I_1 &:= \frac{1}{N_1}\left[-\alpha\sum_{i,k=1}^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(1)}} - \widehat{x_i^{(1)}})^2\right] \le -2\alpha S N_1 M_2(\widehat{V}),\\
I_2 &:= \frac{1}{N_2}\left[-\alpha\sum_{i,k=1}^{N_2}\overline{a}_{k,i}(\widehat{x_k^{(2)}} - \widehat{x_i^{(2)}})^2\right] \le -2\alpha S N_2 M_2(\widehat{W}),\\
I_1 + I_2 &\le -\alpha S\min\{N_1N_2\}\widehat{M_2},\\
I_3 &:= -2\alpha\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\overline{a}_{k,i}(\widehat{x_k^{(2)}} - \widehat{x_i^{(1)}})\widehat{x_i^{(1)}}\frac{1}{N_1} - 2\alpha\sum_{k=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{j,k}(\widehat{x_k^{(1)}} - \widehat{x_j^{(2)}})\widehat{x_j^{(2)}}\frac{1}{N_2}\\
&\le \max\left\{\frac{1}{N_1},\frac{1}{N_2}\right\}2\alpha\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{i,j}(\widehat{x_i^{(1)}} - \widehat{x_j^{(2)}})^2\\
&\le \max\left\{\frac{1}{N_1},\frac{1}{N_2}\right\}2\alpha D\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\widehat{x_i^{(1)}} - \widehat{x_j^{(2)}})^2\\
&= 2\alpha D\max\left\{\frac{1}{N_1},\frac{1}{N_2}\right\}N_1N_2\widehat{M_2}\\
&= 2\alpha D\zeta\widehat{M_2},\\
I_4 &:= -2\alpha\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\overline{a}_{k,i}(x_c^{(2)} - x_c^{(1)})\widehat{x_i^{(1)}}\frac{1}{N_1} - 2\alpha\sum_{k=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{j,k}(x_c^{(1)} - x_c^{(2)})\widehat{x_j^{(2)}}\frac{1}{N_2}\\
&\le 2\alpha D\zeta|x_c^{(1)} - x_c^{(2)}|\sqrt{\widehat{M_2}},\\
I_5 &:= 2\delta\sum_{i=1}^{N_1}\widehat{x_i^{(1)}}x_i^{(1)}(1-(x_i^{(1)})^2)\frac{1}{N_1},\\
I_6 &:= 2\delta\sum_{i=1}^{N_2}\widehat{x_i^{(2)}}x_i^{(2)}(1-(x_i^{(2)})^2)\frac{1}{N_2}.
\end{aligned} \tag{59}$$

Using $x_i^{(1)} = \widehat{x_i^{(1)}} + x_c^{(1)}$ and $\sum_i \widehat{x_i^{(1)}} = 0$, we obtain

$$\begin{aligned} I_5 &= \frac{2\delta}{N_1}\sum_i^{N_1}(1-(x_i^{(1)})^2)\widehat{x_i^{(1)}}^2 + \frac{2\delta}{N_1}\sum_i^{N_1} x_c^{(1)}\widehat{x_i^{(1)}} \\ &= 2\delta M_2(\widehat{V}) - (-\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^2\widehat{x_i^{(1)}}^2 - \frac{2\delta}{N_1}(x_i^{(1)})^2 x_c^{(1)}\widehat{x_i^{(1)}} \\ &= 2\delta M_2(\widehat{V}) - (-\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^2\widehat{x_i^{(1)}}^2 \\ &\leq 2\delta M_2(\widehat{V}). \end{aligned} \tag{60}$$

The last inequality is based on

$$\begin{aligned} \sum_i^{N_1}(x_i^{(1)})^2\widehat{x_i^{(1)}}^2 &= \sum_i^{N_1}(x_i^{(1)})^2((x_i^{(1)})^2 - x_c^{(1)}\widehat{x_i^{(1)}}) \\ &= \frac{1}{2}\sum_i^{N_1}(x_i^{(1)})^2((x_i^{(1)})^2 - (x_c^{(1)})^2 + (x_i^{(1)} - x_c^{(1)})^2) \\ &\geq \frac{1}{2}\sum_i^{N_1}(x_i^{(1)})^2((x_i^{(1)})^2 - (x_c^{(1)})^2) \\ &= \frac{1}{2}\sum_i^{N_1}(x_i^{(1)})^4 - \frac{1}{2}\sum_i^{N_1}(x_i^{(1)})^2((x_c^{(1)})^2 \\ &\geq \frac{1}{2}\sum_i^{N_1}(x_i^{(1)})^4 - \frac{1}{2N_1}\left(\sum_i^{N_1}(x_i^{(1)})^2\right)^2 \geq 0. \end{aligned}$$

Similarly on I_6, one has $I_6 \leq 2\delta M_2(\widehat{W})$. Thus, $I_5 + I_6 \leq 2\delta\widehat{M}_2$. ■

C Experiments

The code for the experiments is available at:

`https://anonymous.4open.science/r/ACMP-092A`

We will replace this anonymous link with a non-anonymous GitHub link after the acceptance. We implement all experiments in Python 3.8.13 with PyTorch Geometric on one NVIDIA ® Tesla A100 GPU with 6,912 CUDA cores and 80GB HBM2 mounted on an HPC cluster.

In addition, we take the official implementation of the Graph Neural Diffusion (GRAND) as diffusion term in (7) from the repository: `https://github.com/twitter-research/graph-neural-pde`

C.1 Details for Experiments

Datasets We consider two types of datasets: Homophilic and Heterophilic. They are differentiated by the *homophily level* of a graph [57]:

$$\mathcal{H} = \frac{1}{|V|}\sum_{v\in V}\frac{\text{Number of } v\text{'s neighbors who have the same label as } v}{\text{Number of } v\text{'s neighbors}}.$$

In the experiments, we have used six homophilic datasets, including Cora [45], Citeseer [63] and Pubmed [51], Computer and Photo [51], and CoauthorCS [64], and three heterophilic datasets: Cornell, Texas and Wisconsin from the WebKB dataset[4]. For completeness, we list the numbers of

[4] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

classes, features, nodes and edges of each dataset, and their homophily level in Table 3. The low homophily level means that the dataset is more heterophilic when most of neighbours are not in the same class, and the high homophily level indicates that the dataset close to homophilic when similar nodes tent to be connected. The datasets we used in Table 3 covers various homophily levels.

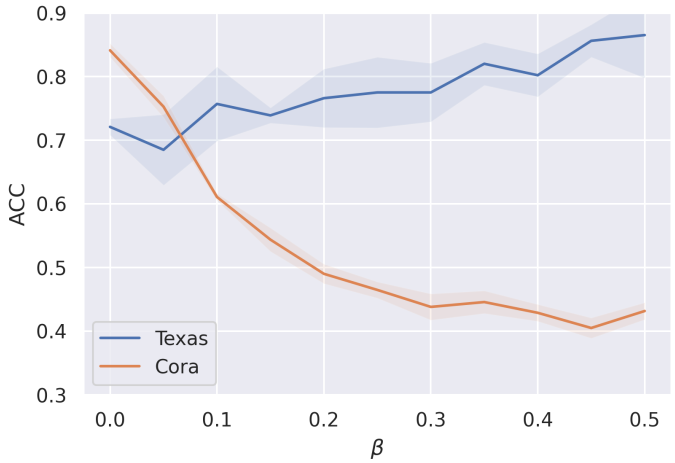

Figure 4: Significance plot for β in terms of test accuracy on Cora (orange) and Texas (blue) with 10 fixed random splits. Cora and Texas belong to homophilic and heterophilic respectively.

Table 3: Information for Graph Datasets Used in Experiments

Dataset	Classes	Features	#Nodes	Edges	Homophily level
Cora	7	1433	2485	5069	0.83
CiteSeer	6	3703	2120	3679	0.71
PubMed	3	500	19717	44324	0.79
CoauthorCS	15	6805	18333	81894	0.80
Computer	10	767	13381	245778	0.77
Photo	8	745	7487	119043	0.83
Texas	5	1703	183	309	0.11
Wisconsin	5	1703	183	499	0.21
Cornell	5	1703	183	499	0.30

Experiment setup For homophilic datasets, we use 10 random weight initializations and 100 random splits, which contains 1,000 tests. Each combination randomly select 20 numbers for each class. For heterophilic data, we use the original fixed 10 split datasets. We fine-tune our model within hyper-parameter search space, as detailed in Table 4. We use the Dormand–Prince adaptive step size scheme (DOPRI5) as the neural ODE solver for all datasets. Hyperparameter search used Ray Tune with a hundred trials using an asynchronous hyperband scheduler with a grace period of 50 epochs. All the details to reproduce our results have been included in the submission and will be publicly available after publication.

C.2 Performance of ACMP to hyperparameter β

Hyperparameter β is the key to introduce the repulsive force in GNN, meaning that when $a_{ij} - \beta$ is negative, the two nodes repel one another. To illustrate β's impact on different datasets, we use GCN as a diffusion term as a_{ij} do not change during the ODE process and all the changes are related to β. As shown by Figure 4, ACMP performs best in Cora (orange curve) when all nodes are attracted to one another i.e. all $a_{ij} - \beta$ is positive. As the beta increases, the performance of the model degrades. In contrast, for the Texas dataset, when all force is attractive, ACMP achieves only 70% accuracy (blue curve). As β increases, most $a_{ij} - \beta$ is negative, and the model's performance gets better. When

Table 4: Hyperparameter Search Space

Hyperparameters	Search Space	Distribution
learning rate	$[10^{-6}, 10^{-1}]$	log-uniform
weight decay	$[10^{-3}, 10^{-1}]$	log-uniform
dropout rate	$[0.1, 0.8]$	uniform
hidden dim	$\{64, 128, 256\}$	categorical
time (T)	$[2, 25]$	uniform
β	$[0, 1]$	uniform

all the force is repulsive, ACMP achieves highest accuracy on Texas datasets, which is in accordance with our claim that repulsive force is important for heterophilic datasets.

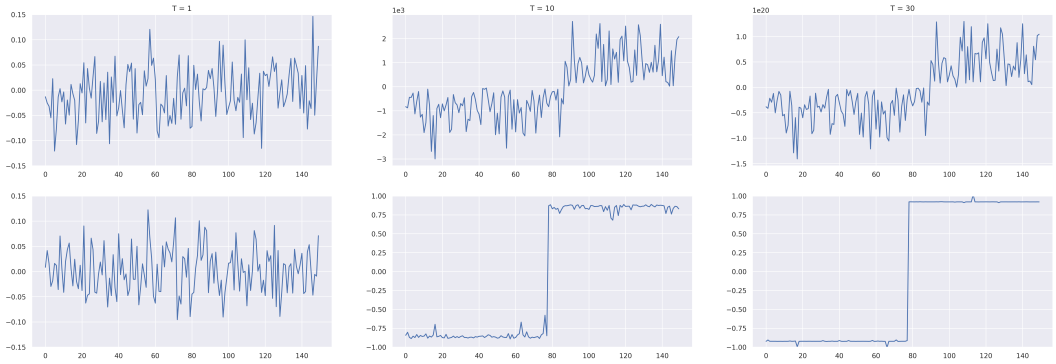

Figure 5: Example of how adding Allen-Cahn terms can prevent the nodes feature from becoming infinite. We choose the first channel in the node's feature of dimension 150. In the first row, the repulsive force is added to message passing without the Allen-Cahn term, and in the second row, the Allen-Cahn term is added to message passing. The first, second and third columns show the neural ODE's initial state, and the states when $T = 10$ and $T = 30$.

C.3 Ablation study for ACMP

Message Passing Performance vs Depths We compare ACMP with various GNN models such as GRAND, GCN, GAT, and GraphSage with different depths on the planetoid datasets. Table 5 lists the nodes classification accuracy on Cora, Citeseer and Pubmed. We observe that ACMP can maintain its model performance as the network deepens and achieve top test accuracy among all listed models using the same depth. ACMP can thus overcome the oversmoothing.

The Allen-Cahn term We now show in Figure 5 how the Allen-Cahn term can stabilize training and prevent node features from blowing up. The first row is the evolution of the diffusion equation without Allen-Cahn term while the second row has the Allen-Cahn term added. We can observe that introducing the repulsive term is essential for bounding GNN outputs, particularly when learning heterophilic datasets. However, naively adding β to message passing will result in all node's features becoming infinite. In the first row of Figure 5 when the Allen-Cahn term is not incorporated, the node's features have increased to 3×10^3 when $T = 10$, from 0.1 when $T = 1$. By the time T equals 30, the node's largest feature becomes 1×10^{20}, which the neural ODE solver and message passing can hardly handle numerically corrected. When we introduce the Allen-Cahn term, the system contains two strong attractors of ± 1, and the nodes are attracted to the two ends of 1 and -1 by their own features.

Table 5: Test Accuracy of Models with Different Depth

Model	depth	Cora	CiteSeer	PubMed
GRAND-1	4	82.80 ± 1.62	73.87 ± 2.12	78.71 ± 1.19
	16	82.75 ± 1.17	72.61 ± 2.42	78.79 ± 0.93
	32	82.19 ± 1.73	72.65 ± 3.15	78.70 ± 1.08
GCN	4	81.35 ± 1.27	70.54 ± 6.61	77.15 ± 3.00
	16	19.70 ± 7.06	24.78 ± 1.45	41.36 ± 1.77
	32	21.86 ± 6.09	24.23 ± 1.65	40.66 ± 1.86
GAT	4	80.95 ± 2.28	72.31 ± 2.82	77.37 ± 1.32
	16	29.14 ± 1.02	24.84 ± 1.45	39.21 ± 0.43
	32	29.75 ± 1.57	24.83 ± 1.45	39.02 ± 0.12
GraphSage	4	79.83 ± 2.43	50.00 ± 14.27	76.01 ± 2.35
	16	25.52 ± 6.45	24.84 ± 1.45	37.55 ± 3.92
	32	29.14 ± 1.02	28.38 ± 2.54	39.21 ± 4.39
ACMP (ours)	4	$\mathbf{83.87} \pm 0.5$	$\mathbf{74.61} \pm 1.04$	$\mathbf{79.74} \pm 0.24$
	16	$\mathbf{83.19} \pm 0.6$	$\mathbf{73.13} \pm 0.85$	$\mathbf{79.16} \pm 0.36$
	32	$\mathbf{83.11} \pm 0.81$	$\mathbf{72.76} \pm 1.05$	$\mathbf{79.81} \pm 1.61$