ACMP: Allen-Cahn Message Passing with Attractive and Repulsive Forces for Graph Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural message passing is a basic feature extraction unit for graph-structured data considering 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 Allen-Cahn force arising in the modeling of phase transition. The dynamics of the system is a reaction-diffusion process which can separate particles without blowing up. This induces an Allen-Cahn message passing (ACMP) for graph neural networks where the numerical iteration for the particle system solution constitutes the message passing propagation. ACMP which has a simple implementation with a neural ODE solver can propel the network depth up to one hundred of layers with theoretically proven strictly positive lower bound of the Dirichlet energy. It thus provides a deep model of GNNs circumventing the common GNN problem of oversmoothing. GNNs with ACMP achieve state of the art performance for real-world node classification tasks on both homophilic and heterophilic datasets.

1 Introduction

Graph neural networks (GNNs) have 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 [4, 6, 12, 13, 28, 60]. Neural message passing [30] serves 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 [16] and coupled oscillator system [36] have been used to define graph convolution that enables 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 the particle system based message passing is similar to that of collective behaviors common in nature and human society [24, 8, 37, 58]. For example, insects form swarms to work; birds form flocks to immigrate; humans form parties to express public opinions. Various mathematical models have been proposed to interpret these behaviors [1, 44, 15, 53, 23]. 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.

Our model is based on an interacting particle system with attractive/repulsive structure. It can be viewed as the graph version of the bi-cluster Cucker-Smale swarming model introduced in [27]. This naturally induces a GNN model to simulate the message passing dynamics. There are two major

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

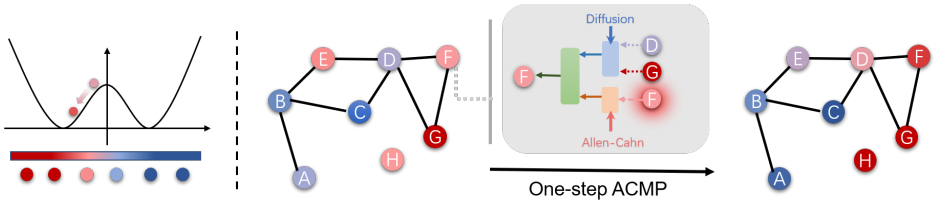

Figure 1: The left figure shows the double-well potential field $W(x)$ and particles $\{x_i\}$ according to their position gain their own potential energy. Here, colors of the nodes illustrate features and correspond to the color bar. Under the potential W, nodes with shallow blue or shallow red have the tendency to turn dark blue or red. But usually, they are influenced by both potential and neighborhood interaction. The right figure shows one-step ACMP propagator. 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' from D, besides its own potential. Resultantly, F will then turn into red in the next layer.

components in this model. First, the co-existence of the attractive and repulsive forces allows particles to separate into two clusters. Second, Allen-Cahn [2] term (or Rayleigh friction [54]), prevents the Dirichlet energy in the evolution from becoming unbounded. These force terms provide the key mechanism to overcome the oversmoothing obstacle of deep networks. Specifically, we will prove that under suitable conditions on the parameters, the dynamics of the particle system will time-asymptotically form 2^d 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 GNNs with ACMP remains lower bounded and the node features are upper bounded, due to introducing Allen-Cahn double-well potential term in the particle equation; 3) the adaption of the proposed GNNs for homophilic and heterophlic node classification tasks can be adjusted flexibly by training. The model can then reach an acceptable trade-off on self-features and neighbor effect, as shown in Figure 1

2 Background

Message Passing in Graph Neural Networks Graph neural networks are a kind of deep neural networks which take graph data as input. Neural Message Passing (MP) [30, 7] 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 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 [35], GAT [59] and GIN [62] 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

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

The MP framework was also developed as PDE solvers in [11] by embedding differential equations as a parameter into message passing like [10]. This paper regards particle PDE evolution as message passing propagation, and the appropriate design of the particle system offers desired properties for the resulting GNN.

Graph neural diffusion Diffusion equations on graphs are defined by [16] under the diffusion equation framework. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $|\mathcal{V}| = N$ nodes and $|\mathcal{E}| = E$ edges. For simplicity, the node features are assumed to be scalar-valued and denoted by x_i of the ith node and set $\mathbf{x} = \oplus x_i$. The diffusion equation on the graph is defined as

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathrm{div}[\mathbf{G}(\mathbf{x}(t), t)\nabla\mathbf{x}(t)] \tag{4}$$

in Section 3 in [16]. Here $\mathbf{G} = \mathrm{diag}(a(x_i(t), x_j(t), t))$ and a is some function on the similarity between nodes i and j. We assume $a = a(x_i, x_j)$ here for notation simplicity while a can depend on time t.

3 Motivations

3.1 Attractive and repulsive force

Despite GRAND generally builds on the diffusion framework, the equation (4) itself can be interpreted in a formulation different from diffusion. In this paper, we analyze the equations of the similar construction with (4) as an *interacting particle system.* Rewrite (4) into a component form, we get

$$\frac{\partial}{\partial t}x_i(t) = \sum_{j\in\mathcal{N}_i} a(x_i, x_j)(x_j - x_i). \tag{5}$$

In the formulation of particle systems, one can easily discover the the evolution trend of the features. If $a(x_i, x_j) > 0$, the direction of x_i's velocity is towards x_j, which means that x_i is attracted by x_j. In the contrast, if $a(x_i, x_j) < 0$, x_i has a trend to take away from x_j. $a(x_i, x_j)$ here, implies the attractive or repulsive force between x_i and x_j. In the GRAND model, all the $a(x_i, x_j)$'s are positive, which means that all the node features in one connected component attract each other. If the weight matrix $(a(x_i, x_j))_{N\times N}$ is right-stochastic, one can prove the convex hull of the features will not dilate in time (see [45, 16]). Such feature aggregation means that message propagates along the edges of the graph and some potential consensus forms in the process.

However, the message propagation does not limit to *consensus* (corresponding to diffusion in GRAND). Information interaction can derive polarization of final judgement when *negative* message matters in some problems instead of *positive* message. For instance, in a node classification task on a bipartite, the neighbour message is negative since connected nodes belong to different classes. In the formulation of particle systems, the mechanism of positive and negative messages can be modelled by adding bias $\beta_{i,j}$ into (5)

$$\frac{\partial}{\partial t}x_i(t) = \sum_{j\in\mathcal{N}_i} (a(x_i, x_j) - \beta_{i,j})(x_j - x_i). \tag{6}$$

The coefficient term $a(x_i, x_j) - \beta_{i,j}$ implies the interactive force. By adjusting $\beta_{i,j}$, there are both attractive and repulsive forces in the system. If $a(x_i, x_j) - \beta_{i,j} > 0$, x_i is attracted by x_j. But if $a(x_i, x_j) - \beta_{i,j} < 0$, x_i is repelled by x_j. If the coefficient equates zero, there is no interaction between x_i and x_j. Then, the dynamics is enabled to adapt both positive and negative message passing. Hence, it adapts both homophilic and heterophilic datasets, which is discussed in Section 6.

3.2 Pseudo-Ginzburg-Landau energy

However, the artificial term of repulsive force may cause the particles separate infinitely. If a couple of repulsive interaction exists for a long time, it is possible that the system will blow up. Correspondingly, the Dirichlet energy in the evolution becomes unbounded. To add the stability of the system, we design a damping term $\delta x_i(1 - x_i^2)$, and we call it Allen-Cahn term. Here, the coefficient $\alpha > 0$ is multiplied just for technical convenience.

$$\frac{\partial}{\partial t}x_i(t) = \alpha \sum_{j\in\mathcal{N}_i} (a(x_i, x_j) - \beta_{i,j})(x_j - x_i) + \delta x_i(1 - x_i^2). \tag{7}$$

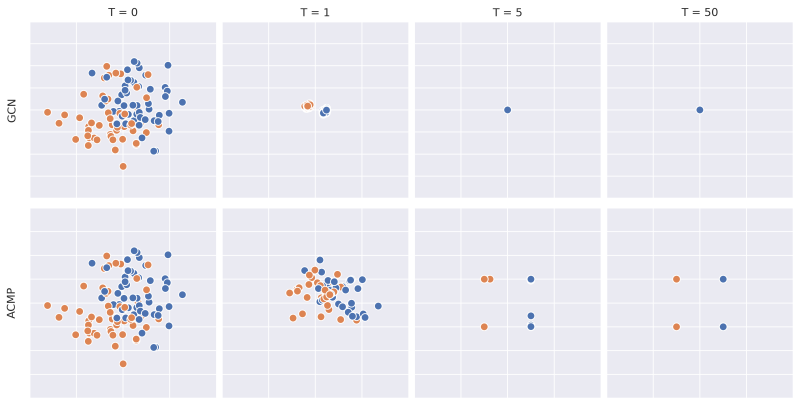

Figure 2: We compare the evolution of node features in GCN and ACMP. We show GCN in the first row and ACMP in the second row. The initial position is represented by the 2-dimensional position of the nodes, which is shown in the first column. The GCN aggregates all node features by taking the weighted average of its neighbors' 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.

Gradient Flow The intuition is to attach an external potential to damp the system. 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 (5) can be characterized by looking into the corresponding Euler-Lagrange equation of the Dirichlet energy.

Let adjacent matrix $\mathbf{A}$ represent the undirected connectivity between nodes x_i and x_j, with $a_{i,j} = 1$ for $(i,j) \in \mathcal{E}$ and $a_{i,j} = 0$ for $(i,j) \notin \mathcal{E}$. The Dirichlet energy $\mathbf{E}$ in terms of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and node features $\mathbf{x} \in \mathbb{R}^{N \times d}$ takes the form

$$\mathbf{E}(\mathbf{x}) = \frac{1}{N} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} a_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2. \tag{8}$$

By calculus of variation, we can formulate the particle equation

$$\frac{\partial \mathbf{x}}{\partial t} = -\nabla_{\mathbf{x}} \mathbf{E}, \quad \frac{\partial x_i}{\partial t} = -\frac{\partial \mathbf{E}}{\partial x_i} = \frac{2}{N} \sum_{j \in \mathcal{N}_i} a_{i,j}(x_j - x_i). \tag{9}$$

On the RHS of (9), the summation takes over the one-hop neighbors $\mathcal{N}_i$ of node i, which aggregates the impact from the neighboring nodes. Equation (9) is (7) when one takes adjacent matrix $\mathbf{A}$ as the weight matrix $(a(x_i, x_j))_{N \times N}$.

Particle equation with Allen-Cahn potential To avoid blowing-up, one can design an external potential which equates infinity at infinity. Here, we define the *pseudo-Ginzburg-Landau energy* on graph $\mathcal{G}$ denoted by $\Phi : L^2(\mathcal{V}) \to \mathbb{R}$, as a combination of the *interacting energy* and double-well potential $W : \mathbb{R} \to \mathbb{R}_+$, $W(x) = (\delta/4)(1 - \|x\|^2)^2$, $\Phi(\mathbf{x}) = \frac{1}{2}\alpha \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} (a_{i,j} - \beta_{i,j}) \|x_i - x_j\|^2 + \sum_{i \in \mathcal{V}} W(x_i)$, with parameters $\alpha, \delta > 0$ to balance two types of energy. From now on, $a_{i,j}$ denotes $a(x_i, x_j)$ for simplicity. Note the pseudo-Ginzburg-Landau energy is not a true energy because the matrix $(a_{i,j} - \beta_{i,j})_{N \times N}$ can be non-positive definite. If $\beta_{i,j}$ all equate zero, it is the Ginzburg-Landau energy defined in [40]. Using this combined energy, we can also derive the Allen-Cahn equation (with repulsion) on the graph $\mathcal{G}$ as

$$\frac{\partial \mathbf{x}}{\partial t} = -\nabla_{\mathbf{x}} \Phi, \quad \frac{\partial x_i}{\partial t} = -\frac{\partial \Phi}{\partial x_i} = \alpha \sum_{j \in \mathcal{N}_i} (a_{i,j} - \beta_{i,j})(x_j - x_i) + \delta x_i (1 - x_i^2). \tag{10}$$

4 Dirichlet Energy

The dynamics (7) can overcome the oversmoothing phenomenon [48, 50, 36]. Oversmoothing phenomenon means that all node features converge to the same constant – consensus forms – as the network deepens, thus the Dirichlet energy will decay to zero exponentially. This idea was first introduced in [14]. [36] gives an explicit form for oversmoothing. 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. In addition, the system will not blow up thanks to the Allen-Cahn term. We put all the proofs and some related supplementary results in the appendix.

Propsition 1 *If $\delta > 0$, the node features $\mathbf{x}_i$ in (7) 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 $\lambda_{\max}$.*

In the following propositions, we imitate the emergent behavior analysis in [27] (see Appendix for details). 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 $\{\mathbf{x}_i^{(2)}\}_{j=1}^{N_2}$ satisfying

$$\begin{aligned} &(i)\ \sup_{0\leq t<\infty} \max_{1\leq i,j\in N_1} |x_i^{(1)}(t) - x_j^{(1)}(t)| < \infty, \quad \sup_{0\leq t<\infty} \max_{1\leq i,j\in N_2} |x_i^{(2)}(t) - x_j^{(2)}(t)| < \infty; \\ &(ii)\ \exists\, C, T^{**} > 0 \textit{ such that } \min_{1\leq i\in N_1, 1\leq j\in N_2} \left\{|x_i^{(1)}(t) - x_j^{(2)}(t)|\right\} \geq C, \quad \forall t > T^{**}, \end{aligned} \tag{11}$$

where $x_i^{(1)}, x_i^{(2)}$ denote any component of $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}$.

We now show the long-time behaviour of model (15) following the analysis of [27] for strength coupling (α, δ) that satisfies the following condition: there exists $\{\beta_{i,j}\}$ such that $\mathcal{I} := \{1, \ldots, N\}$ can be divided into two disjoint groups $\mathcal{I}_1, \mathcal{I}_2$ with N_1 and N_2 particles respectively:

$$\begin{aligned} 0 < S \leq \overline{a}_{i,j} \ &\text{ with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} && \text{for } i,j \in \mathcal{I}_1, \\ 0 < S \leq \overline{a}_{i,j} \ &\text{ with } \overline{a}_{i,j} := a_{i,j} - \beta_{i,j} && \text{for } i,j \in \mathcal{I}_2, \\ 0 \leq \overline{a}_{i,j} \leq D \ &\text{ with } \overline{a}_{i,j} := -(a_{i,j} - \beta_{i,j}) && \text{otherwise}, \end{aligned} \tag{12}$$

where S, D are independent of time t. The S and D in (12) are the repulsive and attractive forces. We show that if the repulsive force between the particles is stronger than the attractive force, that is, $S > D$, the system is guaranteed to have bi-cluster flocking, as shown in Proposition 2 below. 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 $\mathbf{x}_{c,k}^{(s)}(t)$, $k = 1, \ldots, d$, be the kth (dimension) component of the feature $\mathbf{x}_c^{(s)}(t)$, $s = 1, 2$.

Propsition 2 *The system (15) has a bi-cluster flocking if for each $k = 1, \ldots, d$, the initial $|\mathbf{x}_{c,k}^{(1)}(0) - \mathbf{x}_{c,k}^{(2)}(0)| \gg 1$, and if there exists a positive constant η such that*

$$\alpha(S - D)\min\{N_1, N_2\} \geq \delta + \eta, \tag{13}$$

where the δ is the weight factor for the double-well potential in the equation (15).

Propsition 3 *For system (15) 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 Allen-Cahn Message Passing

We propose an *Allen-Cahn Message Passing* (ACMP) neural network based on equation (10), which learns the node features evolution. To our best knowledge, this is the *first time* to introduce a type of message passing to amplify the difference between connected nodes by repulsive force.

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

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

where $\frac{\partial \mathbf{X}(t)}{\partial t}$ is estimated by ACMP defined on $\mathcal{G}$ based on (10). The node features $\mathbf{x}(T)$ at the ending time are fed into an MLP based classifier. Specifically, ACMP reads

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha} \odot \sum_{j \in \mathcal{N}_i} (a(\mathbf{x}_i(t), \mathbf{x}_j(t)) - \beta)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \boldsymbol{\delta} \odot \mathbf{x}_i(t) \odot (1 - \mathbf{x}_i(t) \odot \mathbf{x}_i(t)). \quad (14)$$

Here $\boldsymbol{\alpha}, \boldsymbol{\delta} \in \mathbb{R}^d$ are learnable vectors of the same length as the node feature $\mathbf{x}_i$. We simplify the parameters $\beta_{i,j}$ to one hyper-parameter β, while we can use a more general case when each edge (i, j) uses different $\beta_{i,j}$. The operations of all terms are channel-wise, involving d channels, except $a(\mathbf{x}_i(t), \mathbf{x}_j(t))$, and $\odot$ represents channel-wise multiplication for d feature channels. Figure 1 illustrates the ACMP mechanism (14): The colors of the nodes imply the interaction between connected nodes and itself potential. The resultant action domains one-step ACMP. We derive two typical types of ACMP in terms of how to represent $a(\mathbf{x}_i(t), \mathbf{x}_j(t))$ based on (14).

ACMP-GCN: this is the model with deterministic $a(\mathbf{x}_i(t), \mathbf{x}_j(t))$, which at time $t = 0$, is given by the prior adjacency matrix of the graph,

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha} \odot \sum_{j \in \mathcal{N}_i} (a_{i,j}^{\text{GCN}} - \beta)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \boldsymbol{\delta} \odot \mathbf{x}_i(t) \odot (1 - \mathbf{x}_i(t) \odot \mathbf{x}_i(t)), \quad (15)$$

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

ACMP-GAT: we can replace $a_{i,j}^{\text{GCN}}$ in (15) by the attention coefficients (3) of GAT, which contain extra trainable parameters to incorporate to measue the similarity information between two nodes. The system evolution then drives the edges to update in each iteration of message passing.

One can adjust $\beta \in \mathbb{R}^+ \cup \{0\}$ such that the attractive force and repulsive force both present in the graph to enrich the message passing effect. Besides, if one chooses $\boldsymbol{\delta} = 0$, $\beta = 0$ in ACMP-GCN or ACMP-GAT in (15), our model is reduced to the graph neural diffusion network (GRAND) in [16]. In experiments, we would make significant use of nontrivial $\boldsymbol{\delta}$ and β.

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 [18, 38, 47, 16]. 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. Here, the number of runs of our message passing, or the *network depth* of ACMP-GNN is equal to the numerical iteration number n_t set in the solver. We leave the details about numerical solver to Appendix.

Computational Complexity The computational complexity of the ACMP is $\mathcal{O}(NEdn_t)$, where n_t, N, E and d are number of time steps in time interval $[0, T]$, number of nodes, number of edges and number of feature dimension, respectively. Since our model only considers nearest (one-hop) neighbors, E is significantly smaller than that of graph rewiring [29, 3] and multi-hop [64] methods.

Channel mixer The previous methods ACMP-GCN and ACMP-GAT are demonstrated channel-wise. However, channel mixing can be spontaneously introduced from the perspective of diffusion coefficients. Specifically, suppose $\mathbf{x}_i \in \mathbb{R}^d$ in (14), then scalar coefficient $a(\mathbf{x}_i(t), \mathbf{x}_j(t)) \in \mathbb{R}$ or vector coefficient $a(\mathbf{x}_i(t), \mathbf{x}_j(t)) \in \mathbb{R}^d$ correspond to operations without channel mixing, while, tensor coefficients $\mathbf{a}(\mathbf{x}_i(t), \mathbf{x}_j(t)) \in \mathbb{R}^{d\times d}$ correspond to a learnable channel mixer such as ACMP-GAT realization $\mathbf{a}^{\mathrm{attn}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - \beta$, where $\mathbf{a}^{\mathrm{attn}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) \in \mathbb{R}^{d\times d}$ and $\beta \in \mathbb{R}$. The channel mixer can be introduced by generalizing the Dirichlet energy to high dimension, for example,

$$\mathbf{E}(\mathbf{x}) := \frac{1}{N}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}(\mathbf{x}_i - \mathbf{x}_j)^T a_{i,j}(\mathbf{x}_i - \mathbf{x}_j),$$

where $\mathbf{a}_{i,j} \in \mathbb{R}^{d\times d}$ are connectivity tensors, which mix the features in different channels.

6 Experiments

6.1 Dirichlet Energy

We first illustrate the evolution of the Dirichlet energy of ACMP by an 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 [35], GAT [59], GRAND [16] 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' features to collapse to the same value. For ACMP, the energy stabilizes at the level that relies upon the roots of the Allen-Cahn potential in (14) after slightly decaying in the first two layers.

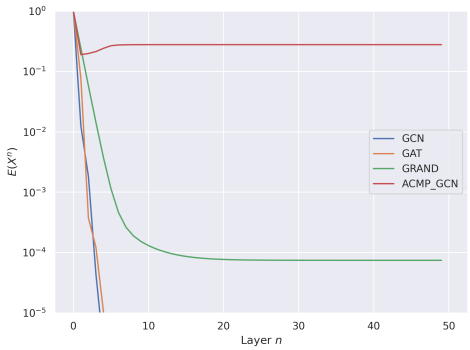

Figure 3: Evolution of Dirichlet energy $\mathbf{E}(\mathbf{X}^n)$ of layer-wise node features $\mathbf{X}^n$ propagated through GCN, GAT, GRAND and ACMP-GCN.

Homophily level	Texas 0.11	Wisconsin 0.21	Cornell 0.30
GPRGNN [19]	78.4 ± 4.4	82.9 ± 4.2	80.3 ± 8.1
H2GCN [64]	$\mathbf{84.9 \pm 7.2}$	$\mathbf{87.7 \pm 5.0}$	$\mathbf{82.7 \pm 5.3}$
GCNII [17]	77.6 ± 3.8	80.4 ± 3.4	77.9 ± 3.8
Geom-GCN [51]	66.8 ± 2.7	64.5 ± 3.7	60.5 ± 3.7
PairNorm [63]	60.3 ± 4.3	48.4 ± 6.1	58.9 ± 3.2
GraphSAGE [33]	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 [59]	52.2 ± 6.6	49.4 ± 4.1	61.9 ± 5.1
GCN [35]	55.1 ± 5.2	51.8 ± 3.1	60.5 ± 5.3
GraphCON [36]	$\mathbf{85.4 \pm 4.2}$	$\mathbf{87.8 \pm 3.3}$	$\mathbf{84.3 \pm 4.8}$
ACMP-GCN (ours)	$\mathbf{86.2 \pm 3.0}$	$\mathbf{86.1 \pm 4.0}$	$\mathbf{85.4 \pm 7.0}$

Table 1: Node classification results on heterophilic datasets. We use the 10 fixed splits for training, validation and test from [51] and show the mean and standard deviation of test accuracy. We show the best three methods 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 datasets. Graph data is considered as *homophilic* if its homophily level [51] 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 [41], Citeseer [56] and Pubmed [46]. Moreover, we evaluate our model on the Amazon co-purchasing graphs Computer and Photo [46], and CoauthorCS [57]. We compare our model with traditional GNN models: Graph Convolutional Network (GCN) [35], Graph Attention Network (GAT) [59], Mixture Model Networks [43] and GraphSage [33]. We also compare our results with recent ODE-based GNNs, Continuous Graph Neural Networks (CGNN) [61], Graph Neural Ordinary Differential Equations (GDE) [52] and Graph Neural Diffusion (GRAND) [16]. To address the limitations of this evaluation methodology proposed by [57], we report results for all datasets using 100 random splits with 10 random initialization's, 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 [35]	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 [59]	81.8 ± 1.3	71.4 ± 1.9	78.7 ± 2.3	90.5 ± 0.6	78.0	85.7
GAT-ppr [59]	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 [43]	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 [33]	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 [33]	76.6 ± 1.9	67.5 ± 2.3	76.1 ± 2.3	85.0 ± 1.1	N/A	90.4 ± 1.3
CGNN [61]	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 [52]	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 [16]	$\mathbf{83.6 \pm 1.0}$	$\mathbf{73.4 \pm 0.5}$	$\mathbf{78.8 \pm 1.7}$	$\mathbf{92.9 \pm 0.4}$	$\mathbf{83.7 \pm 1.2}$	$\mathbf{92.3 \pm 0.9}$
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[1]. 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 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 most nodes are repelled by their neighbors. Several GNNs exploiting multi-hop information can achieve high performance in node classification [64, 39]. However, high-order neighbor information will make the adjacency matrix dense and therefore can not be extended to large graphs, due to heavier computational cost. In our model, we take only one-hop information into account and add repulsive force ($\beta \geq 0$) to message passing, which has achieved the same or higher level of accuracy as multi-hop models in heterophilic datasets.

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

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

7 Related work

Neural differential equations The topic of neural ODEs becomes an emerging field since E [25] and Chen et al.'s work [18], with many follow-up works. Neural ODEs have been introduced into the GNN field: [5] used continuous residual modules for graph kernels; [52] extended the framework of GNN to continuous time. [55] applied Hamiltonian mechanics to graph networks to predict future states. GRAND [16] approached graph deep learning as a continuous diffusion process and propagated GNNs by the graph diffusion equation. [26] combined diffusion and wave PDEs for GNNs, and GraphCON [36] 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 [21, 22, 20, 45, 15] with asymptotic estimates [45, 27, 31, 32, 34]. The flocking problem is to some degree similar to the general consensus problem [49] which studies the emergent behaviours for multi-agent systems. The Cucker-Smale model [21] is a famous model in this field considering a second-order system adopting to classical dynamics.

Allen-Cahn based variational graph models In [9, 40, 42] and references therein, authors extended Allen-Cahn related potential to graphical framework and developed a class of variational algorithms to solve the clustering, semisupervised learning and graph cutting problems. The new ingredient of graph neural network which enables us to combine learnable attraction and repulsion separates our method from the classical variational graph models.

8 Conclusion

We develop a new message passing method with simple implementation. The method is based on the Allen-Cahn particle system with repulsive force. The proposed ACMP inherits the characteristic dynamics of the particle system and thus shows adaption for node classification tasks with high homophily difficulty. Also, it propels networks to dozens of layers without getting oversmoothing. A strictly positive lower bound of the Dirichlet energy is shown by theoretical and experimental results which guarantees non-oversmoothing of ACMP. Experiments show excellent performance of the model for various real datasets.

Limitations First, ACMP is less interpretable for link prediction and graph-level prediction tasks. Second, we limited our attention to the double-well potential W. It is possible to develop learnable alternatives corresponding to the datasets. Also, in theory, the parameter β for the adjacency matrix is learnable. In practice, however, it is not easy to train and we make it as a hyperparameter to be tuned. In addition, although ACMP is computed efficiently, it relies on the neural ODE solver which is not stable when the message passing has hundreds of runs.

References

[1] 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.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Andrea L Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.

[10] 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.

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

[12] 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.

[13] 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.

[14] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. In *ICML Workshop on Graph Representation Learning*, 2020.

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

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

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

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

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

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

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

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

[23] 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.

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

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

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

[27] 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.

[28] 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.

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

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

[31] 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.

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

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

[34] 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.

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

[36] T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-coupled oscillator networks. In *ICML*, 2022.

[37] 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.

[38] 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.

[39] 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.

[40] Xiyang Luo and Andrea L Bertozzi. Convergence of the graph Allen-Cahn scheme. *Journal of Statistical Physics*, 167(3):934–958, 2017.

[41] 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.

[42] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An MBO scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6(4):1903–1930, 2013.

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

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

[49] 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.

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

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

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

[53] 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.

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

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

[56] 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.

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

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

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

[60] 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.

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

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

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

[64] 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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   (b) Did you describe the limitations of your work? Yes
   (c) Did you discuss any potential negative societal impacts of your work? No,we believe our work has no negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? Yes
   (b) Did you include complete proofs of all theoretical results? Yes
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes
   (b) Did you mention the license of the assets? NA
   (c) Did you include any new assets either in the supplemental material or as a URL? Yes
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? NA
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? NA
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? NA
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

A Model variants

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 (15), particles could escape from one well of the potential of W to another well. We can circumvent this instability by enhancing the attraction of the wells, which can be achieved by reducing the diffusion power around wells:

$$\frac{\partial}{\partial t}\mathbf{x}_i(t) = \boldsymbol{\alpha}\odot\sum_{j\in\mathcal{N}_i}(a^{\mathrm{GNN}}(\mathbf{x}_i(t),\mathbf{x}_j(t))-\beta)(\mathbf{x}_j(t)-\mathbf{x}_i(t))\left(1-\mathbf{x}_i(t)^{\odot 2}\right)^{\odot 2}+\boldsymbol{\delta}\odot\mathbf{x}_i(t)\odot\left(1-\mathbf{x}_i(t)^{\odot 2}\right). \tag{16}$$

where 'GNN' in a^{GNN} can be GCN or attn, and $z^{\odot 2}$ is $z\odot z$. With this modification in (16), in any channel k, if any particle $\mathbf{x}_i^{(k)}$ gets caught in one potential well, then it is not likely to escape:

Propsition 4 *For (16), there exists a proper $\delta' > 0$ such that $\mathbf{x}_i^{(k)} \in [-1, -1+\delta') \cup (1-\delta', 1]$, then particle $\mathbf{x}_i^{(k)}$ cannot transition into another well.*

Proof. For the $\beta = 0$ case, assume $x_i = -1+\epsilon$ for $\epsilon \le \delta' < 1$ 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 (28), the above inequality is equivalent to

$$\sum_{j\in\mathcal{N}_i} a_{i,j}x_j < \frac{\delta}{\alpha}\frac{1-\epsilon}{2-\epsilon}\frac{1}{\epsilon}+\epsilon-1 \le \frac{\delta}{2\alpha}\frac{1}{\epsilon}+\epsilon-1 \le \frac{\delta}{2\alpha\epsilon}. \tag{17}$$

Since $\{x_j\}_{j=1}^N$ are bounded (See Proposition 1.), (17) 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 $\epsilon \le \delta' < 1$ at a certain time t_0. Similarly with (17), we have

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

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

B Supplementary Results and 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) [16],

$$\dot{x}_i = \alpha\sum_{j:(i,j)\in\mathcal{E}} a_{i,j}(x_j - x_i). \tag{18}$$

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 $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{\max} <$

∞. 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

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

Since the Dirichlet energy can also be written as

$$\mathbf{E}(\mathbf{x}(t)) = \mathbf{x}(t)^\top \mathcal{L}\mathbf{x}(t), \tag{20}$$

Taking (19) to (20) gives

$$\mathbf{E}(\mathbf{x}(t)) = \mathbf{x}(0)^\top \mathbf{U}^\top e^{-\mathbf{\Lambda} t}\mathbf{\Lambda} e^{-\mathbf{\Lambda} t}\mathbf{U}\mathbf{x}(0). \tag{21}$$

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

Propsition 5 *We also consider a more general case,*

$$\frac{d}{dt}x_i(t) = \sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i), \tag{22}$$

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 (22), we obtain $\mathrm{d}x_c/\mathrm{d}t = 0$ for any $t > 0$. Without loss of generality, we may assume

$$x_c(0) = 0, \tag{23}$$

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 (22) 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 (22) and sum over x_i to obtain*

$$x_i \frac{dx_i}{dt} = \sum_{j\in N(j)} a\left(x_i, x_j\right)\left(x_j - x_i\right) x_i \tag{24}$$

$$\Rightarrow \quad \frac{d}{dt}\|x\|^2 = -2\sum_{(i,j)\in\mathcal{E}} a\left(x_i, x_j\right)\left(x_j - x_i\right)^2 \tag{25}$$

$$\Rightarrow \quad \frac{d}{dt}\|x\|^2 \leqslant -2a_{\min}\sum_{(i,j)\in\mathcal{E}} \left(x_j - x_i\right)^2 \tag{26}$$

The RHS in (26) 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, (26) leads to

$$\frac{d}{dt}\|x\|^2 \leqslant -2a_{\min}\lambda_{\min}\|x\|^2. \tag{27}$$

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 Allen-Cahn term

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

$$\begin{cases} \frac{d}{dt}x_i(t) = \alpha \sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i) + \delta x_i \left(1 - x_i^2\right), \\ a(x_i, x_j) = a(x_j, x_i) \geq 0, \quad \forall i, j \in \mathcal{V} \quad \sum_i a(x_i, x_j) = 1, \forall j \in \mathcal{V}. \end{cases} \tag{28}$$

For (28), we can assert that the stable equilibrium of any x_i is limited in the interval $[-1, 1]$.

Propsition 6 *Suppose* $\mathbf{x}^* = (x_1^*, \ldots, x_N^*)$ *is a global equilibrium (or steady state solution) of (28) on* $\mathbb{R}$ *and* $\mathbf{x}$*, then* $x_i^* \in [-1, 1]$.

Proof. Suppose x^* achieves the equilibrium of (28), 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 (28). As a simple transference of Lemma 3.2 in [31], we can prove this. Set

$$x^M(t) := \max_i x_i(t), \quad x^m(t) := \min_i x_i(t), \tag{29}$$

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 7 *Let* $\{x_i\}$ *be the solutions of (28), 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 [31]. 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,

$$\begin{aligned} \frac{dx_i}{dt}(t^*) &= \alpha \sum_j a(x_j, x_i)(x_j(t^*) - x_i(t^*)) + \delta x_i(t^*)(1 - x_i^2(t^*)) \\ &= \alpha \sum_j a(x_j, x_i) x_j(t^*) \\ &\geq 0. \end{aligned} \tag{30}$$

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 (16). We can prove that if any particle x_i gets caught in one potential well, then it will not escape from that well.

B.3 The Attractive-repulsive Model

We first show that the solution features of graph in Allen-Cahn model below is bounded. For simplicity of the proof, we rewrite (15) in component form where we let $a(x_i, x_j) := a_{i,j} - \beta_{i,j}$:

$$\frac{d}{dt}x_i(t) = \alpha \sum_{j:(i,j)\in\mathcal{E}} a(x_i, x_j)(x_j - x_i) + \delta x_i \left(1 - x_i^2\right). \tag{31}$$

Model (31) allows negative $a(x_i, x_j)$ which is different from the condition in (28).

Proof of Proposition 1. We multiply x_i on both sides of the following equation and sum over x_i to obtain

$$\begin{aligned}
&\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i) - x_i^3 + x_i \\
\Rightarrow &\frac{1}{2}\frac{\mathrm{d}x_i^2}{\mathrm{d}t} = \sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)\,x_i - x_i^4 + x_i^2 \\
\Rightarrow &\frac{1}{2}\sum_{i\in\mathcal{V}} \frac{\mathrm{d}x_i^2}{\mathrm{d}t} = -\sum_{i\in\mathcal{V}} \left(\sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)\,x_i - x_i^4 + x_i^2 \right).
\end{aligned} \tag{32}$$

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

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|x\|^2 = -\frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} a(x_i, x_j)\,(x_j - x_i)^2 - \sum_{i\in\mathcal{V}} x_i^4 + \|x\|^2. \tag{33}$$

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 (33) 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 1.

■

Recall (12) and rewritte it as

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

For the attractive-repulsive model (34), we can refer to the the proof of its Theorem 5.1. in [27].

We define the following notations for further proof:

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

Remark 1 *(13) indicates that the repulsive force between the particles should be stronger than the attractive force($S > D$).*

To prove Proposition 2, 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*

$$
\begin{aligned}
\frac{d}{dt}M_2 = &-\frac{\alpha}{N_1}\sum_{i,k}^{N_1}\overline{a}_{k,i}(x_k^{(1)} - x_i^{(1)})^2 - \frac{2\alpha}{N_1}\sum_{k}^{N_2}\sum_{i}^{N_1}\overline{a}_{i,k}(x_k^{(2)} - x_i^{(1)})x_i^{(1)}\\
&+ \frac{2\delta}{N_1}\sum_{i}^{N_1}(x_i^{(1)})^2(1-(x_i^{(1)})^2)\\
&- \frac{\alpha}{N_2}\sum_{j,k}^{N_2}\overline{a}_{k,j}(x_k^{(2)} - x_j^{(2)})^2 - \frac{2\alpha}{N_2}\sum_{j}^{N_2}\sum_{i}^{N_2}\overline{a}_{j,k}(x_k^{(1)} - x_j^{(2)})x_j^{(2)}\\
&+ \frac{2\delta}{N_2}\sum_{i}^{N_2}(x_j^{(2)})^2(1-(x_j^{(2)})^2).
\end{aligned}
\tag{35}
$$

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

$$\sup_{0\leq t<\infty} M_2(t) \leq M_2^\infty < \infty. \tag{36}$$

Lemma 2 *Let $\{x_i\}$ be a solution to (34) with $\delta > 0$. Then $\widehat{M_2}$ satisfies*

$$\frac{d}{dt}\widehat{M_2} \leq -2\eta\widehat{M_2} + 2\alpha D\zeta|x_c^{(1)} - x_c^{(2)}|\sqrt{\widehat{M_2}}, \tag{37}$$

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

Proof of Proposition 2

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

$$\begin{aligned} |x_c^{(1)} - x_c^{(2)}| &= \left| \frac{1}{N_1} \sum_{i=1}^{N_1} x_i^{(1)} - \frac{1}{N_2} \sum_{i=1}^{N_2} x_i^{(2)} \right| \\ &\leq \frac{1}{N_1} \sum_{i=1}^{N_1} |x_i^{(1)}| + \frac{1}{N_2} \sum_{i=1}^{N_2} |x_i^{(2)}| \\ &\leq 2\sqrt{\frac{1}{N_1} \sum_{i=1}^{N_1} (x_i^{(1)})^2 + \frac{1}{N_2} \sum_{i=1}^{N_2} (x_i^{(2)})^2} \\ &= 2\sqrt{M_2(t)} \leq 2\sqrt{M_2^\infty}. \end{aligned} \tag{38}$$

(b) (Uniform boundedness of $\widehat{M}_2$) By Lemma 2 and (38),

$$\begin{aligned} \frac{d}{dt}\sqrt{\widehat{M}_2} &\leq -\eta\sqrt{\widehat{M}_2} + \alpha D\zeta |x_c^{(1)} - x_c^{(2)}| \\ &\leq -\eta\sqrt{\widehat{M}_2} + 2D\alpha\zeta\sqrt{M_2^\infty}. \end{aligned} \tag{39}$$

Use Gronwall's lemma to obtain

$$\begin{aligned} \sqrt{\widehat{M}_2(t)} &\leq \sqrt{\widehat{M}_2(0)} e^{-\eta t} + \frac{2D\alpha\zeta\sqrt{M_2^\infty}}{\eta}(1 - e^{-\eta t}) \\ &\leq \max\left\{ \sqrt{\widehat{M}_2(0)}, \frac{2D\alpha\zeta\sqrt{M_2^\infty}}{\eta} \right\} := C_3. \end{aligned} \tag{40}$$

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

$$\begin{aligned} \frac{d}{dt}|x_c^{(1)} - x_c^{(2)}|^2 =& 2(x_c^{(1)} - x_c^{(2)}) \frac{\alpha}{N_1} \sum_k^{N_2} \sum_i^{N_1} \overline{a}_{k,i} (\widehat{x_k^{(2)}} + x_c^{(2)} - \widehat{x_i^{(1)}} - x_c^{(1)}) \\ &- 2(x_c^{(1)} - x_c^{(2)}) \frac{\alpha}{N_2} \sum_k^{N_2} \sum_i^{N_1} \overline{a}_{k,i} (\widehat{x_k^{(2)}} + x_c^{(2)} - \widehat{x_i^{(1)}} - x_c^{(1)}) \\ &- \frac{2\delta}{N_1} \sum_i^{N_1} (x_i^{(1)})^3 (x_c^{(1)} - x_c^{(2)}) + \frac{2\delta}{N_2} \sum_i^{N_2} (x_i^{(2)})^3 (x_c^{(1)} - x_c^{(2)}) \\ \geq& 2\alpha \left(\frac{1}{N_1} + \frac{1}{N_2} \right) \sum_i^{N_1} \sum_j^{N_2} \overline{a}_{i,j} (x_c^{(1)} - x_c^{(2)})^2 + \mathcal{I}_{c1} + \mathcal{I}_{c2}. \end{aligned} \tag{41}$$

By the Cauchy-Schwarz inequality,

$$\begin{aligned} |\mathcal{I}_{c1}| &= -2\alpha \left(\frac{1}{N_1} + \frac{1}{N_2} \right) \sum_i^{N_1} \sum_j^{N_2} \overline{a}_{i,j} (\widehat{x_j^{(2)}} - \widehat{x_i^{(1)}})(x_c^{(1)} - x_c^{(2)}) \\ &\leq 2\alpha \left(\frac{1}{N_1} + \frac{1}{N_2} \right) D\sqrt{N_1 N_2} |x_c^{(1)} - x_c^{(2)}| \sqrt{\sum_{i,j}^{N_1,N_2} (\widehat{x_j^{(2)}} - \widehat{x_i^{(1)}})^2} \\ &= 2\alpha \left(\frac{1}{N_1} + \frac{1}{N_2} \right) D N_1 N_2 |x_c^{(1)} - x_c^{(2)}| \widehat{M}_2. \end{aligned} \tag{42}$$

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

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

Then, one gets

$$\begin{aligned}\mathcal{I}_{c2} &\ge -2\left|x_c^{(1)}-x_c^{(2)}\right|\left|\frac{\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^3 + \frac{\delta}{N_2}\sum_i^{N_2}(x_i^{(2)})^3\right| \\ &\ge -2\left|x_c^{(1)}-x_c^{(2)}\right|\delta\sqrt{\max\{N_1,N_2\}}M_2(t)^{\frac{3}{2}}.\end{aligned} \tag{43}$$

Hence,

$$\begin{aligned}\frac{d}{dt}|x_c^{(1)}-x_c^{(2)}|^2 \ge &\left[\left(2\alpha(\frac{1}{N_1}+\frac{1}{N_2})\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\overline{a}_{i,j}\right)+\delta\right]|x_c^{(1)}-x_c^{(2)}|^2 \\ &-\alpha D(N_1+N_2)\sqrt{\widehat{M}_2}-\delta\sqrt{\max\{N_1,N_2\}}M_2^{\frac{3}{2}}.\end{aligned} \tag{44}$$

Combining with Lemma 1 and (40), one obtains the estimate

$$\begin{aligned}\frac{d}{dt}|x_c^{(1)}-x_c^{(2)}| \ge &\left(2\alpha\left(\frac{1}{N_1}+\frac{1}{N_2}\right)\sum_i^{N_1}\sum_j^{N_2}\overline{a}_{i,j}+\delta\right)|x_c^{(1)}-x_c^{(2)}| \\ &-\alpha D(N_1+N_2)C_3-\delta\sqrt{\max\{N_1,N_2\}}(M_2^\infty)^{\frac{3}{2}}.\end{aligned} \tag{45}$$

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

$$|x_c^{(1)}(0)-x_c^{(2)}(0)| \ge \frac{\alpha D(N_1+N_2)C_3+\delta\sqrt{\max\{N_1,N_2\}}(M_2^\infty)^{\frac{3}{2}}}{\delta} := \frac{C_4}{\delta}, \tag{46}$$

then,

$$|x_c^{(1)}(t)-x_c^{(2)}(t)| \ge \frac{C_4}{\delta}+(|v_c(0)-w_c(0)|-\frac{C_4}{\delta})e^{\delta t} \ge \frac{C_4}{\delta}. \tag{47}$$

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

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

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

$$|x_i^{(1)}(t)-x_j^{(2)}(t)| \ge C > 0, \quad \forall i,j. \tag{48}$$

Combing with Proposition 1, we finish the proof. ■

Remark 2 *The proof of Proposition 3 is included in part (d) of proof of Proposition 2.*

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:

$$\begin{aligned}
\mathbf{E}(x) &= \frac{1}{N}\sum_{i,j} a_{i,j}(x_i - x_j)^2 \\
&= \frac{1}{N}\left[\sum_{i,j\in\mathcal{I}_1} a_{i,j}(x_i^{(1)} - x_j^{(1)})^2 + \sum_{i,j\in\mathcal{I}_2} a_{i,j}(x_i^{(2)} - x_j^{(2)})^2 + \sum_{i\in\mathcal{I}_1, j\in\mathcal{I}_2} a_{i,j}(x_i^{(1)} - x_j^{(2)})^2\right] \\
&\geq \frac{1}{N}\sum_{i\in\mathcal{I}_1, j\in\mathcal{I}_2} a_{i,j}(x_i^{(1)} - x_j^{(2)})^2 \\
&\geq \frac{C^2\eta_2}{N}.
\end{aligned} \tag{49}$$

Proof of Lemma 1.

$$\begin{aligned}
\frac{d}{dt}M_2(V) &= \frac{2}{N_1}\sum_{i=1}^{N_1} x_i^{(1)}\dot{x_i^{(1)}} \\
&= -\frac{\alpha}{N_1}\sum_{i,k}^{N_1} \overline{a}_{k,i}(x_k^{(1)} - x_i^{(1)})^2 - \frac{2\alpha}{N_1}\sum_{k}^{N_2}\sum_{i}^{N_1} \overline{a}_{i,k}(x_k^{(2)} - x_i^{(1)})x_i^{(1)} \\
&\quad + \frac{2\delta}{N_1}\sum_{i}^{N_1}(x_i^{(1)})^2(1 - (x_i^{(1)})^2).
\end{aligned} \tag{50}$$

Similarly,

$$\begin{aligned}
\frac{d}{dt}M_2(W) &= \frac{2}{N_2}\sum_{i=1}^{N_2} x_i^{(2)}\dot{x_i^{(2)}} \\
&= -\frac{\alpha}{N_2}\sum_{j,k}^{N_2} \overline{a}_{k,j}(x_k^{(2)} - x_j^{(2)})^2 - \frac{2\alpha}{N_2}\sum_{j}^{N_2}\sum_{i}^{N_2} \overline{a}_{j,k}(x_k^{(1)} - x_j^{(2)})x_j^{(2)} \\
&\quad + \frac{2\delta}{N_2}\sum_{i}^{N_2}(x_j^{(2)})^2(1 - (x_j^{(2)})^2).
\end{aligned} \tag{51}$$

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

$$\begin{aligned}
\frac{d}{dt}M_2 \leq& \frac{D\alpha}{N_1}\sum_{k=1}^{N_2}\sum_{i=1}^{N_1}\left((x_k^{(2)} - x_i^{(1)})^2 + (x_i^{(1)})^2\right) + \frac{D\alpha}{N_2}\sum_{k=1}^{N_1}\sum_{j=1}^{N_2}\left((x_k^{(1)} - x_j^{(2)})^2 + (x_j^{(2)})^2\right) \\
&+ \frac{2\delta}{N_1}\sum_{i=1}^{N_1}(x_i^{(1)})^2(1 - (x_i^{(1)})^2) + \frac{2\delta}{N_2}\sum_{i=1}^{N_2}(x_i^{(2)})^2(1 - (x_i^{(2)})^2).
\end{aligned} \tag{52}$$

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 (52) yield a Riccati-type differential inequality:

$$\begin{aligned}
\frac{d}{dt}M_2 \leq& 2D\alpha N_2 M_2(W) + 3D\alpha N_2 M_2(V) + 2D\alpha N_1 M_2(V) + 3D\alpha N_2 M_2(W) \\
&+ 2\delta M_2 - \delta(M_2)^2 \\
\leq& (\alpha C_m + 2\delta)M_2 - \delta(M_2)^2.
\end{aligned} \tag{53}$$

Let y be a solution of the following ODE:

$$y' = \alpha C_m y - \delta y^2. \tag{54}$$

Then, the solution $y(t)$ to equation 54 satisfies

$$M_2(t) \le y(t) \le \max\left\{\frac{\alpha C_m}{\delta} + 2, M_2(0)\right\} =: M_2^\infty. \tag{55}$$

■

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,

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

Combine the two equations,

$$\frac{d}{dt}\widehat{M_2} = \sum_{i=1}^{6} I_i, \tag{56}$$

where

$$
\begin{aligned}
I_1 &:= \frac{1}{N_1}\left[-\alpha \sum_{i,k=1}^{N_1} \overline{a}_{k,i}(\widehat{x_k^{(1)}} - \widehat{x_i^{(1)}})^2\right] \le -2\alpha S N_1 M_2(\widehat{V}),\\
I_2 &:= \frac{1}{N_2}\left[-\alpha \sum_{i,k=1}^{N_2} \overline{a}_{k,i}(\widehat{x_k^{(2)}} - \widehat{x_i^{(2)}})^2\right] \le -2\alpha S N_2 M_2(\widehat{W}),\\
I_1 + I_2 &\le -\alpha S \min\{N_1 N_2\}\widehat{M_2},\\
I_3 &:= -2\alpha \sum_{k=1}^{N_2}\sum_{i=1}^{N_1} \overline{a}_{k,i}(\widehat{x_k^{(2)}} - \widehat{x_i^{(1)}})\widehat{x_i^{(1)}}\frac{1}{N_1} - 2\alpha \sum_{k=1}^{N_1}\sum_{j=1}^{N_2} \overline{a}_{j,k}(\widehat{x_k^{(1)}} - \widehat{x_j^{(2)}})\widehat{x_j^{(2)}}\frac{1}{N_2}\\
&\le \max\left\{\frac{1}{N_1}, \frac{1}{N_2}\right\} 2\alpha \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \overline{a}_{i,j}(\widehat{x_i^{(1)}} - \widehat{x_j^{(2)}})^2\\
&\le \max\left\{\frac{1}{N_1}, \frac{1}{N_2}\right\} 2\alpha D \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} (\widehat{x_i^{(1)}} - \widehat{x_j^{(2)}})^2\\
&= 2\alpha D \max\left\{\frac{1}{N_1}, \frac{1}{N_2}\right\} N_1 N_2 \widehat{M_2}\\
&= 2\alpha D \zeta \widehat{M_2},\\
I_4 &:= -2\alpha \sum_{k=1}^{N_2}\sum_{i=1}^{N_1} \overline{a}_{k,i}(x_c^{(2)} - x_c^{(1)})\widehat{x_i^{(1)}}\frac{1}{N_1} - 2\alpha \sum_{k=1}^{N_1}\sum_{j=1}^{N_2} \overline{a}_{j,k}(x_c^{(1)} - x_c^{(2)})\widehat{x_j^{(2)}}\frac{1}{N_2}\\
&\le 2\alpha D \zeta |x_c^{(1)} - x_c^{(2)}|\sqrt{\widehat{M_2}},\\
I_5 &:= 2\delta \sum_{i=1}^{N_1} \widehat{x_i^{(1)}} x_i^{(1)}(1 - (x_i^{(1)})^2)\frac{1}{N_1},\\
I_6 &:= 2\delta \sum_{i=1}^{N_2} \widehat{x_i^{(2)}} x_i^{(2)}(1 - (x_i^{(2)})^2)\frac{1}{N_2}.
\end{aligned}
\tag{57}
$$

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

$$
\begin{aligned}
I_5 &= \frac{2\delta}{N_1}\sum_i^{N_1}(1 - (x_i^{(1)})^2)\widehat{x_i^{(1)}}^2 + \frac{2\delta}{N_1}\sum_i^{N_1} x_c^{(1)}\widehat{x_i^{(1)}}\\
&= 2\delta M_2(\widehat{V}) - (-\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^2\widehat{x_i^{(1)}}^2 - \frac{2\delta}{N_1}(x_i^{(1)})^2 x_c^{(1)}\widehat{x_i^{(1)}}\\
&= 2\delta M_2(\widehat{V}) - (-\frac{2\delta}{N_1}\sum_i^{N_1}(x_i^{(1)})^2\widehat{x_i^{(1)}}^2\\
&\le 2\delta M_2(\widehat{V}).
\end{aligned}
\tag{58}
$$

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 (14) 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 [51]:

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

In the experiments, we have used six homophilic datasets, including Cora [41], Citeseer [56] and Pubmed [46], Computer and Photo [46], and CoauthorCS [57], and three heterophilic datasets: Cornell, Texas and Wisconsin from the WebKB dataset[2]. For completeness, we list the numbers of 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.

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.

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

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

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

C.2 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.

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
	64	80.87 ± 2.28	$\mathbf{69.84} \pm 2.66$	NA
	128	77.22 ± 2.88	NA	NA
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	$\mathbf{83.87} \pm 0.52$	$\mathbf{74.61} \pm 1.04$	$\mathbf{79.74} \pm 0.24$
	16	$\mathbf{83.19} \pm 0.67$	$\mathbf{73.13} \pm 0.85$	$\mathbf{79.16} \pm 0.36$
	32	$\mathbf{83.11} \pm 0.81$	$\mathbf{72.76} \pm 1.05$	$\mathbf{79.81} \pm 1.61$
	64	$\mathbf{80.48} \pm 1.21$	68.92 ± 1.37	$\mathbf{78.01} \pm 0.01$
	128	$\mathbf{80.30} \pm 1.18$	$\mathbf{67.83} \pm 0.02$	$\mathbf{77.98} \pm 0.01$

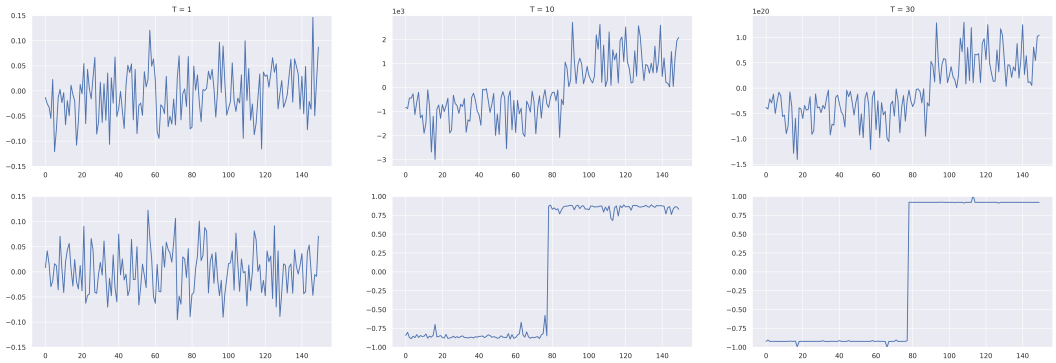

Figure 5: Example of how adding Allen-Cahn terms can prevent the nodes feature from becoming infinite. We choose the first channel in the node's feature of dimension 150. In the first row, the repulsive force is added to message passing without Allen-Cahn term, and in the second row, Allen-Cahn term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

Allen-Cahn term We now show in Figure 5 how 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 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 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 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.