Postprint: A Game-Theoretic Community Detection Algorithm Based on Node Attributes

Authors: Zhang Xiankun, Ren Jing, Liu Yuanbo, Su Jing

Date: 2018-11-29T00:00:00+00:00

Abstract

In recent years, the mining and discovery of high-quality communities has become a hot topic in social network research. This paper proposes a game-theoretic community detection algorithm based on node attributes, G_{NA} (game algorithm based on node attributes for community detection). It views the process of community detection as a game among nodes in the network, where the game ends when no node can improve its own payoff. First, G_{NA} proposes a payoff function based on node degree attributes; then, during the iterative process, nodes are sorted in descending order of importance and sequentially choose strategies to improve their payoff; finally, comparative experiments are conducted between the proposed algorithm and existing algorithms on different real-world and artificial networks, and the results show that the proposed algorithm outperforms other algorithms.

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers
Accepted Paper

Game Algorithm Based on Node Attributes for Community Detection

Zhang Xiankun, Ren Jing, Liu Yuanbo, Su Jing
(School of Computer Science & Information Engineering, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract: In recent years, the mining and detection of high-quality communities has become a hot topic in social network research. This paper proposes a game algorithm based on node attributes for community detection (G_{NA}). The community detection process is modeled as a game among nodes in the network, which terminates when all nodes can no longer improve their own utilities. First, G_{NA} proposes a utility function based on node degree attributes. Then, during the iteration process, nodes are sorted by importance in descending order and sequentially select strategies to improve their utilities. Finally, the proposed algorithm is compared with existing algorithms on various real-world and artificial networks, and the results demonstrate its superiority.

Keywords: community detection; game; node attributes; utility function

0 Introduction

Complex networks serve as an abstraction of complex systems, with many real-world systems describable and analyzable through complex network properties. In such representations, graph nodes denote individual entities and edges represent relationships between them [1~3]. Community structure constitutes a universal characteristic of complex networks, where the entire network comprises numerous communities with dense intra-community connections and sparse inter-community links. When the intersection of node sets from any two communities is empty, they are non-overlapping communities; otherwise, they are overlapping communities. Research on complex networks has remained a focal point across numerous domains, including protein structure analysis, urban road construction, and social marketing. The emergence of social platforms like Weibo and Facebook has further intensified the importance of studying community structures in complex networks.

Current community detection algorithms primarily include graph partitioning methods such as spectral bisection and the K-L algorithm; divisive approaches like the GN algorithm; and modularity-based methods such as simulated annealing and extremal optimization. The K-L algorithm requires pre-specifying the sizes of two communities, severely limiting its practical applicability. Spectral bisection can only partition two communities at a time, necessitating multiple iterations for multiple communities, which reduces efficiency and accuracy. The GN algorithm is an edge-removal method based on divisive clustering principles, using edge betweenness as a similarity metric. It deletes edges with high betweenness, causing network splits much faster than random edge deletion. However, GN does not know the final number of communities beforehand and may involve repeated shortest-path calculations.

After Newman proposed the modularity Q metric, researchers developed various modularity-based optimization methods that transform community detection into an optimization problem seeking an optimal objective function solution. In 2010, Chen et al. [4] first applied game theory to community detection, proposing a game-theoretic framework. However, their algorithm did not consider node attributes, and random selection of initial nodes introduced significant instability in results. Building upon Chen et al.' s work, this paper proposes the G_{NA} algorithm. First, we introduce a utility function based on node degree attributes as a modification to Chen et al.' s utility function. Second, during iteration, nodes are sorted by importance and sequentially select strategies to improve utilities. Finally, comparative experiments on different real-world and artificial networks demonstrate the superiority of our algorithm.

The main contributions of this paper are: (a) proposing a community detection game algorithm based on node attributes; (b) introducing a utility function incorporating node degree attributes; (c) sorting nodes by importance for strategy selection in the game; and (d) imposing no restrictions on the size or number of communities to be detected.

---

1 Background

1.1 Game Theory Fundamentals

1.1.1 Definition of Game Theory

Game theory [5] is a theory concerning strategic interactions among participants. It considers both predicted and actual behaviors of individuals in games and studies their optimal strategies. When individual interests conflict, each participant' s payoff depends not only on their own actions but also on others' actions, requiring each to respond optimally to others' choices.

1.1.2 Elements of Game Theory

Game theory comprises several elements: players, strategies, payoffs, and equilibrium, with players, strategies, and payoffs being the most fundamental. Players are decision-making entities in the game who maximize their payoffs by selecting strategies; they can be individuals, groups, or nature itself. A player' s strategy set represents their feasible strategies. A strategy profile is an ordered set formed by each participant selecting one strategy.

1.1.3 Nash Equilibrium

Nash equilibrium [6] refers to a strategy profile where, to maximize their own payoff, each participant's strategy must be the best response to others' strategies. As rational participants, no one would deviate from this profile to their own detriment.

Definition 1 (Nash Equilibrium). In a game with n participants, a strategy profile $s^* = (s_1^*, s_2^*, ..., s_n^*)$ is a Nash equilibrium if for every player i, s_i^* is the optimal strategy given other participants' choices, i.e., for all $s_i \in S_i$,

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

Equivalently, s_i^* solves the maximization problem:

$$s_i^* \in \arg\max_{s_i \in S_i} u_i(s_i, s_{-i}^*)$$

Thus, if no participant can unilaterally change their strategy to improve payoff, the strategy profile constitutes a Nash equilibrium.

1.1.4 Potential Games and Potential Functions

Potential games [7] represent a special class of games with an important convergence property. For potential games, there exists a locally linear potential function P that reflects the change in a participant' s utility function when only that participant' s strategy changes. Formally, for each strategy space L and each node' s strategy L'_i, if

$$\omega_i \cdot (u_i(L_i, L_{-i}) - u_i(L'_i, L_{-i})) \geq P(L) - P(L'_i, L_{-i})$$

where ω_i represents participant i' s weight, then the community formation game is a potential game. Nash equilibrium theory states that potential games have at least one Nash equilibrium point, and all local maxima of the potential function are Nash equilibria.

1.2 The Game Algorithm

In 2010, Chen et al. proposed a game-theoretic community detection algorithm, modeling community formation as a community formation game where each network individual is a selfish participant. Each node improves its utility by joining, leaving, or switching communities until the algorithm reaches Nash equilibrium, yielding the final community structure. This framework reflects the organic community formation process in real networks.

The algorithm' s utility function comprises a gain function and a loss function. The gain function is $g_i(L) = (1 - L_i/m)$, and the loss function is $l_i(L) = \frac{1}{2}(1 - L_i/m)$. The basic procedure is: (a) initialize each node as a separate community; (b) randomly select nodes to choose their best strategies; (c) repeat step (b) until Nash equilibrium is reached.

Chen et al.' s algorithm has several shortcomings: (a) the random node selection order for strategy updates causes unstable partitioning results; and (b) it fails to consider how node attributes affect nodes' rights and participation in strategy selection. Building on Chen et al.' s work, many researchers have improved the algorithm, primarily focusing on utility functions and strategy directions. For instance, in 2011, Alvari et al. [8] proposed a utility function based on node similarity and introduced a new similarity measure, though experimental results were unsatisfactory. In 2017, Zhou et al. [9] applied game theory to community detection, proposing a utility function based on node similarity and two node update strategies. However, these existing algorithms do not consider the impact of node attributes on payoff and game sequencing.

---

2 The G_{NA} Algorithm

Based on analysis of existing community detection algorithms, this paper proposes G_{NA}, a game algorithm based on node attributes for community detection. The algorithm models the community partitioning process as a game among nodes. Given a complex network, we assume each node is a selfish participant. Nodes have different importance levels during community formation, determining their strategy selection order and contribution to community payoff. Therefore, we initialize communities using node attributes and allow nodes to select strategies in a certain order to improve payoff. Community structures can be interpreted as equilibrium states of the game. The algorithm allows each node to select multiple communities, enabling overlapping community detection.

Given a static network graph $G = (V, E)$ with $n = |V|$ and $m = |E|$, we assume G is an undirected, unweighted network. Elements in set V are called nodes or agents. Each node selects a subset of communities it wishes to join from all possible community sets $[k] = \{1, 2, ..., k\}$. Notably, the final number of communities may be much smaller than k.

2.1 Utility Function of the Algorithm

In social networks, when a node joins a community to increase its own benefit, it pays a corresponding cost. For example, when an employee switches companies for higher salary, they may face penalty fees from their previous employer. Therefore, node payoff is determined by both gain and loss functions. Based on Newman modularity and incorporating the contribution ratio of node degree values to community payoff, we propose a gain function. Node payoff represents the node' s contribution to modularity.

Based on Newman modularity with node degree value proportion, we propose the gain function for node i as:

$$g_i(L) = \left(1 - \frac{L_i}{m}\right) \cdot \sum_{j \neq i} \left[\frac{1}{2}\hat{p}_{ij} \cdot A_{ij} \cdot \delta(L_i, L_j)\right]$$

where $\delta(L_i, L_j) = 1$ if $|L_i \cap L_j| \geq 1$; otherwise $\delta(L_i, L_j) = 0$. A is the adjacency matrix of G, and $\hat{p}_{ij}$ represents the degree value proportion when node i joins node j' s community, calculated as:

$$\hat{p}_{ij} = \frac{d_i}{\sum_{j' \in C_j} d_{j'}}$$

where C_j denotes the node set of node j' s community, including the newly joined node i.

The loss function is:

$$l_i(L) = \frac{1}{2}\left(1 - \frac{L_i}{m}\right)$$

where m is the number of edges in the network and L_i is node i' s community label.

Therefore, the utility function is:

$$u_i(L) = g_i(L) - l_i(L)$$

As previously discussed, potential games allow pure Nash equilibria. In potential games, an associated potential function $\Phi(\cdot)$ is defined for node strategy selection. If for each strategy space L and each node' s strategy L'_i we have:

$$\Phi(L) - \Phi(L'_i, L_{-i}) = u_i(L_i, L_{-i}) - u_i(L'_i, L_{-i})$$

then the community formation game is a potential game. Clearly, our proposed utility function is linear with respect to 1/2, making the game a potential game that can reach Nash equilibrium.

2.2 Node Strategies

In our community formation game, node v_i' s strategy is a subset of communities it wishes to join, i.e., a subset of $[k]$. We define $L_i \subseteq [k]$ as the strategy or community label of node v_i, allowing $L_i = \emptyset$ (node may belong to no community). Define $L = (L_1, L_2, ..., L_n)$ as the strategy space—a vector of all nodes' community labels.

Let L_{-i} denote community labels of all nodes except i, and (L'_i, L_{-i}) represent the strategy space where the i-th element of L is replaced by L'_i. Define node v_i' s payoff function as $u_i(L) = g_i(L) - l_i(L)$. In the community formation game, given other nodes' strategies L_{-i}, node v_i' s optimal strategy is:

$$L^*_i \in \arg\max_{L'_i \subseteq [k]} u_i(L'_i, L_{-i})$$

Node strategies include: - **Join**: Node adds a new label to L_i to join a community beyond its current ones. - **Leave**: Node removes a label from L_i to leave a community. - **Convert**: Node replaces a label in L_i to switch from one community to another.

These strategies are illustrated in Figures [Figure 1: see original paper]-[Figure 3: see original paper]. In each diagram, dashed lines separate different communities numbered 1 and 2 from left to right.

Figure [Figure 1: see original paper] shows the join strategy. In Figure 1(a), node 1 belongs to community 2 with label $L_1 = \{2\}$. In Figure 1(b), after node 1 joins community 1, its label becomes $L_1 = \{2, 1\}$.

Figure [Figure 2: see original paper] shows the leave strategy. In Figure 2(a), node 1 belongs to communities 1 and 2 with label $L_1 = \{1, 2\}$. In Figure 2(b), after node 1 leaves community 1, its label becomes $L_1 = \{2\}$.

Figure [Figure 3: see original paper] shows the convert strategy. In Figure 3(a), node 1 belongs to community 2 with label $L_1 = \{2\}$. In Figure 3(b), after node 1 converts from community 2 to community 1, its label becomes $L_1 = \{1\}$.

2.3 Algorithm Description

We first calculate each node' s importance and sort nodes by importance values in descending order, then sequentially select strategies to update payoffs.

Node importance measures a node' s influence in the entire network. We adopt Zhang et al.' s [10] Bayesian network-based user node importance calculation method to normalize importance values for all nodes. This importance is based on prior attributes, but normalization based solely on prior attributes is insufficient because nodes are tightly connected—nodes linked to more important nodes become more important themselves. Greater node importance means greater influence on other nodes and higher priority for strategy selection. The importance calculation formula is:

$$NI(i) = Inf(i) + \alpha \sum_{j \in N(i)} \frac{NI(j)}{d_j}$$

where $NI(i)$ denotes node i' s importance, $Inf(i)$ is node i' s prior importance, and α measures the influence coefficient of neighbor importance on node i (0–1). In Zhang et al.' s paper "Label propagation algorithm for community detection based on node importance and label influence," $\alpha = 0.4$ yields optimal results, so we use $\alpha = 0.4$ in our experiments, giving:

$$NI(i) = Inf(i) + 0.4 \sum_{j \in N(i)} \frac{NI(j)}{d_j}$$

where $N(i)$ is node i' s neighbor set and d_j is neighbor node j' s degree.

The G_{NA} algorithm is straightforward: first, initialize each node as a separate community and calculate node importance, sorting nodes in descending order; then set iteration count to 1, and while iterations are fewer than 2,000, nodes sequentially select strategies to update payoffs; finally, output the network partition result and modularity value. The main steps are shown in Algorithm 1.

Algorithm 1: G_{NA} Algorithm Steps
Input: Network $G = (V, E)$
Output: Network community structure and corresponding modularity value

1. Initialize each node as a separate community; initialize payoff $u_i = 0$
2. Calculate each node' s importance value $NI(i)$
3. Sort nodes by importance value in descending order to obtain node sequence
4. Set iteration count $t = 1$
5. **do**
6. For each node in sorted order:
7. `Select strategy and calculate payoff using equation (6)`
8. `Choose the strategy maximizing node payoff and update node's community label`
9. **while** iteration count $t < 2000$

10. Obtain community partition result and modularity value
11. **end**

3 Experimental Analysis

Researchers typically validate community detection algorithms on both real-world and synthetic networks. To verify our algorithm' s performance, we conduct experiments on both types of networks.

3.1 Community Detection on Real-World Networks

3.1.1 Experimental Datasets We evaluate our algorithm on five real-world networks: Zachary' s karate club network (Karate [11]), dolphin network (Dolphins [12]), school friendship network (Friendship [13]), US political books network (Polbooks [14]), and American college football league network (Football [15]). Table shows the number of nodes and edges for each network.

Table Number of nodes and edges of real-world networks

Network	Nodes	Edges
Karate	34	78
Dolphins	62	159
Friendship	69	220
Polbooks	105	441
Football	115	613

From top to bottom, the networks increase in size with more nodes and edges.

3.1.2 Evaluation Metric We use Newman [14] modularity Q to evaluate community detection quality. In 2006, Newman proposed modularity Q to assess community partitioning results for networks with unknown community structures. For each partition, Q measures the difference between actual intra-community edges and expected edges in a random configuration. For unweighted graphs, it can be understood as the sum of intra-community edge degrees minus the total degree of community nodes. The formula is:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j)$$

where A_{ij} is the adjacency matrix (1 if nodes i and j are connected, 0 otherwise), d_i is node i's degree, c_i is node i's community, m is the total number of edges, and $\delta(c_i, c_j)$ is an indicator function (1 if nodes i and j belong to the same community, 0 otherwise). The probability of node j connecting to any node is $d_j/2m$, so the expected edge between nodes i and j in a random case is $d_i d_j/2m$. Q ranges from $-1/2$ to 1, with values closer to 1 indicating more distinct community structure. Q can measure both partition accuracy and algorithm effectiveness.

3.1.3 Experimental Results and Analysis The G_{NA} algorithm' s partitioning results for Karate, Dolphins, Friendship, Polbooks, and Football networks are shown in Figures Figure 4: see original paper–(e).

Figure [Figure 4: see original paper] shows the communities discovered for real-world networks by G_{NA}. As shown in Figure 4(a), G_{NA} partitions the Karate network into 5 communities, discovering 2 overlapping nodes (1 and 34). Figure 4(b) shows the Dolphins network partitioned into 12 communities with 11 overlapping nodes (1, 2, 9, 15, 16, 18, 21, 31, 48, 52, 55). Figure 4(c) shows the Friendship network partitioned into 5 communities with 2 overlapping nodes (15 and 47). Figure 4(d) shows the Polbooks network partitioned into 7 communities with 9 overlapping nodes (10, 11, 13, 15, 21, 48, 68, 85, 100). Figure 4(e) shows the Football network partitioned into 6 communities with 5 overlapping nodes (10, 24, 37, 45, 51). (Overlapping nodes are shown in black.)

G_{NA} discovers richer overlapping structures compared to previous research. By identifying overlapping communities, G_{NA} provides more meaningful information about community structures. The results demonstrate that G_{NA} can discover finer-grained communities than previous methods, revealing more detailed network structures useful for studying community interconnections and larger community structures.

We compare G_{NA} with Game, LPA_{NI}, and LFM [16] algorithms, with results shown in Figure [Figure 5: see original paper].

Figure [Figure 5: see original paper] shows the experimental results for real-world network partitioning. G_{NA} outperforms other algorithms on real-world networks. Moreover, the Game algorithm (game-theoretic) performs better than LPA_{NI} and LFM, indicating that applying game theory to community detection has promising research prospects and significance.

3.2 Community Detection on Artificial Networks

3.2.1 Experimental Datasets We use LFR benchmark networks [17] for synthetic network experiments. LFR networks include parameters: N (number of nodes), k (average degree), maxk (maximum degree), minc (minimum community size), maxc (maximum community size), on (number of overlapping nodes belonging to multiple communities), om (number of communities each overlapping node belongs to), and mu (mixing parameter) representing the probability of internal vs. external connections. As mu increases, community detection becomes more difficult. Table lists LFR network generation parameters.

Table Parameters of LFR networks

We generate three groups of non-overlapping LFR networks with 200, 500, and 1,000 nodes. Each group has mixing parameters of 0.1, 0.2, and 0.3, as shown in Table .

Table Parameters of non-overlapping LFR networks

Network ID	N	k	maxk	minc	maxc	mu
NO_{LFR1}-3	200	30	40	10	30	0.1, 0.2, 0.3
NO_{LFR4}-6	500	30	40	20	50	0.1, 0.2, 0.3
NO_{LFR7}-9	1000	30	50	20	80	0.1, 0.2, 0.3

We also generate two groups of overlapping LFR networks with overlapping node proportions of 2%, 4%, 6%, and 8%. One group has 200 nodes with mu = 0.1, and the other has 1,000 nodes with mu = 0.2, as shown in Table .

Table Parameters of overlapping LFR networks

Network ID	N	k	maxk	minc	maxc	mu	on	om
O_{LFR1}-4	200	30	40	10	30	0.1	4,8,12,16	2
O_{LFR5}-8	1000	30	50	20	80	0.2	20,40,60,80	2

3.2.2 Evaluation Metric We use Normalized Mutual Information (NMI) to evaluate the difference between algorithm-detected communities and ground-truth communities in artificial networks. NMI [17] is defined as:

$$NMI(H^a, H^b) = \frac{I(H^a, H^b)}{\sqrt{H(H^a)H(H^b)}}$$

where H^a and H^b represent community structures, k^a and k^b denote the number of communities, n_h^a is the number of nodes in the h-th community of H^a, and n_{hl} is the number of nodes simultaneously in the h-th community of H^a and the l-th community of H^b. NMI ranges from 0 to 1, with higher values indicating greater consistency between detected and actual community structures.

3.2.3 Experimental Results and Analysis We compare G_{NA} with the Game algorithm on artificial networks. Figure [Figure 6: see original paper] shows results for non-overlapping networks.

Figure [Figure 6: see original paper] shows detection results of non-overlapping networks. Figures 6(a)-(c) present results for networks with 200, 500, and 1,000 nodes respectively. Figure 6(a) shows that when mu = 0.1, G_{NA}' s results are close to Game' s. Figure 6(c) shows that when mu = 0.1, Game slightly outperforms G_{NA}. However, as mu increases, G_{NA} significantly outperforms Game across all three network sizes, demonstrating G_{NA}' s superiority for networks with less distinct community structures.

Figure [Figure 7: see original paper] shows results for overlapping networks.

Figure [Figure 7: see original paper] shows detection results of overlapping networks. Figures 7(a) and (b) present results for networks with N = 200, mu = 0.1 and N = 1,000, mu = 0.2 respectively. The x-axis shows the proportion of overlapping nodes. Figure 7(a) shows G_{NA} outperforms Game. In Figure 7(b), G_{NA} outperforms Game for 0, 20, and 40 overlapping nodes, while performing slightly worse at 60 overlapping nodes but better at 80. Overall, G_{NA} achieves superior results compared to Game.

---

4 Conclusion

This paper proposes G_{NA}, a community detection game algorithm based on node attributes, addressing limitations of existing algorithms. To overcome the limitation of ignoring node attributes' influence on strategy selection, we propose a gain function incorporating node degree proportions. To address the shortcoming of neglecting node attributes' impact on selection order, we introduce a game-theoretic community detection algorithm based on node importance sorting. In this algorithm, nodes are sorted by importance in descending order and sequentially choose strategies (join, leave, or convert communities) to improve payoff until no node can increase its utility.

Future work will further integrate game theory with the essence of community detection to propose more suitable utility functions. Additionally, experimental observations show nodes tend to join communities containing their neighbors, suggesting future work could restrict node strategies to neighbor community selections.

References

[1] Kunegis J. Social network datasets [M]. New York: Springer, 2017.
[2] Yao Ying. Research on complex network community detection technology based on genetic optimization [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2017.
[3] Guo Lei, Xu Xiaoming. Complex network [M]. Shanghai: Shanghai Science and Technology Education Press, 2006.
[4] Wei Chen, Zhenming Liu, Xiaorui Sun, et al. A game-theoretic framework to identify overlapping communities in social networks [J]. Data Mining & Knowledge Discovery, 2010, 21(2): 224-240.
[5] Rubinstein, Ariel. A course in game theory [M]. [S. l.]: MIT Press, 1994.
[6] Aumann R J, Brandenburger A. Epistemic conditions for Nash equilibrium [M]// Readings in Formal Epistemology. [S. l.]: Springer International Publishing, 2016: 113-136.
[7] Monderer D, Shapley L S. Potential games [J]. Games & Economic Behavior,

1996, 14(1): 124-143.
[8] Alvari H, Hashemi S, Hamzeh A. Detecting overlapping communities in social networks by game theory and structural equivalence concept [M]. Berlin: Springer, 2011.
[9] Zhou Xu, Zhao Xiaohui, Liu Yanheng, et al. A game theoretic algorithm to detect overlapping community structure in networks [J]. Physics Letters A, 2018, 382(13): 872-879.
[10] Zhang Xiankun, Ren Jing, Song Chen, et al. Label propagation algorithm for community detection based on node importance and label influence [J]. Physics Letters A, 2017, 381(33): 2691-2698.
[11] Zachary W W. An information flow model for conflict and fission in small groups [J]. Journal of Anthropological Research, 1977, 33(4): 452–473.
[12] Lusseau D, Schneider K, Boisseau O J, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations [J]. Behavioral Ecology and Sociobiology, 2003, 54(4): 396-405.
[13] Yang Liang, Cao Xiaochun, He Dongxiao, et al. Modularity based community detection with deep learning [C]// Proc of International Joint Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2016: 2252-2258.
[14] Newman M E J. Modularity and community structure in networks [J]. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577-8582.
[15] Girvan M, Newman M E J. Community structure in social and biological networks [J]. Proc. Natl Acad. Sci. USA, 2002, 99(12): 7821-7826.
[16] Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure of complex networks [J]. New Journal of Physics, 2008, 11(3): 19-44.
[17] Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities [J]. Physical Review E, 2009, 80(1): 16118.

Note: Figure translations are in progress. See original paper for figures.