CS 229r Spectral Graph Theory in Computer Science, Lecture 14 Oct 20, 2020

Lecture 14

Instructor: Salil Vadhan *Scribe: Kai Wang*

1 Recall: Randomness-efficient Error Reduction Using Expanders

The main goal of having random walks on expanders is that we want to efficiently reduce the error of a randomized algorithm. For example, suppose that we have a randomized algorithm with one-sided error 1/2. That said, we should accept if at least one execution of the randomized algorithm accepts and reject otherwise.

Suppose the randomized algorithm takes n random bits to run[1]. We can independently run the randomized algorithm k times to reduce the error to 2^{-k}. But this requires nk random bits and runtime $O(k)$ in total. What we want to achieve here is to save the number of random bits to $n + O(k)$ while maintaining the same runtime $O(k)$ by the use of random walks on expanders.

We can imagine each vertex as a revelation of n random bits, i.e., $v \in V = \{0,1\}^n$. Given a graph $G = (V, E)$ with $N = |V| = 2^m, V = \{0,1\}^n$. Ideally, we want the following random walks on the expander G efficiently reduce the error to 2^{-k}.

Algorithm 1: Expander Random Walk

```
input: G
initialization: x_1 ~ V = {0,1}^n
for i ∈ {2, 3, ··· , t} do
    x_i ~ {x_{i-1}'s D neighbors}
    Check if the randomized algorithm accepts x_i
output: True if the randomized algorithm ever accepts and False otherwise.
```

This algorithm runs in $O(t)$ steps and only requires $n + tO(\log D)$ random bits, where D is the degree of the regular graph G. If we set $t = O(k)$, then everything is satisfied. The only thing we need to check is whether this reduces the error to 2^{-k}.

Theorem 1. *If G has spectral expansion $\gamma = 1 - \omega$ and $V_1, ..., V_t$ are a random walk on G with uniform start vertex V_1 then:*

$$\forall \text{ bad event } B \subseteq V, \quad Pr\left[\bigcap_{j=1}^{t} (V_j \in B)\right] \leq (\mu + \omega(1-\mu))^t \tag{1}$$

where $\mu = \frac{|B|}{|V|}$.

This theorem implies that if the graph is a good expander (i.e., $\gamma = \Theta(1)$), we know that $\mu + \omega(1-\mu) = \mu + (1-\gamma)(1-\mu) = 1 - \gamma(1-\mu)$ is strictly smaller than 1 if $\mu \neq 1$. This implies that the probability that we always hit the bad event B reduces exponentially. We can simply choose $t = O(\frac{k}{\log \mu + \omega(1-\mu)}) = O(k)$ to reduce the error $(\mu + \omega(1-\mu))^t$ to 2^{-k}.

Remark When the randomized algorithm has two-sided error, we can also reduce the two-sided error by random walks on expanders via Chernoff bound and majority vote/median.

[1] In the textbook, we use m to represent the number of random bits. But to align with the later part of the course, we use n here instead.

Until now we have completed the proof of error reduction of expander random walks. However, there is one premise that we have not covered yet. We do not know whether a good expander always exists. We require to efficiently run random walk on a large expander since the size of the expander can be too large and cannot be efficiently written down. Lastly, we also require the expander to have roughly the same size as $|V| = 2^n$. It is not clear how to efficiently construct an expander with a given order of size.

Remark We can randomly generate the expander when we are doing random walk, but this will require additional randomness when we move along the random walk, which is othorgonal to what we want to achieve. So instead we want a fixed expander but we can still efficiently run random walk without explicitly writting down the expander.

1.1 Review: Matrix Decomposition

Previously, we used the theorem of matrix decomposition to prove the error reduction property of expander random walks. Since it will be used later of the course and it is good to recall it here.

Definition 2 (Spectral Norm)**.**

$$\|M\| = \max_{x \neq 0} \frac{\|Mx\|}{\|x\|} = \textit{ largest eigenvalue of } M \tag{2}$$

Lemma 3 (Matrix decomposition)**.** *G with number of nodes n has spectral expansion γ iff*

$$W = \gamma J + (1-\gamma)E, \tag{3}$$

where J is an all $\frac{1}{n}$ matrix and E is a matrix with $\|E\| \leq 1$.

In particular, we can compute Wv by

$$Wv = \gamma Jv + (1-\gamma)Ev = \gamma v^{\parallel} + (1-\gamma)e \tag{4}$$

where $v^{\parallel}$ is the projection of v in direction of u (uniform distribution) and $\|e\| \leq \|v\|$.

Definition 4 (Vector Decomposition)**.** *We can decompose $v = v^{\parallel} + v^{\perp}$, where $v^{\parallel} \parallel u$ and $v^{\perp} \perp u$.*

In particular, we have

$$Wv = Wv^{\parallel} + Wv^{\perp} = v^{\parallel} + e \qquad \text{s.t. } \|e\| \leq (1-\gamma)\left\|v^{\perp}\right\| \text{ and } e \perp u \tag{5}$$

Clearly we can see that matrix decomposition and vector decomposition create different estimate of the term Wv. Each of these decompositions has its own advantage, e.g., matrix decomposition creates a convex combination, while vector decomposition maintains the othorgonality.

2 Explicit Construction of Expanders

As we previously mentioned, our goal is to efficiently construct an expander roughly with a given size.

Goal: construct an infinite family of graphs $\mathcal{G} = \{G_i\}$ such that

1. $\exists$ constant d s.t. every G_i is d-regular.
2. $\exists$ constant $\gamma > 0$ s.t. every G_i has spectral expansion at least γ.
3. (Fully explicit) given $a \in \{1, 2, \cdots, n_i\}$ and $j \in \{1, 2, \cdots, d\}$ where n_i is the number of vertices in G_i, we can compute the j-th neighbor of vertex a in G_i in time poly$(\log n_i)$.

4. The family $\{n_i\}$ of sizes is not too spare, i.e., we want to convert the family of expanders into a family of all sizes without losing too much expansion.

Example 5 (Construction of Mildly Explicit Expanders)**.** *Given a prime p, we can construct a graph with p vertices from an undirected p-cycle. We connect $x \in \mathbb{Z}_p \backslash \{-1, 0, 1\}$ to its inverse x^{-1} in $\mathbb{Z}_p$. Lastly, we add self-loops to $\{-1, 0, 1\}$. This graph is a 3-regular graph with good expansion (proof relies on number theory).*

The reason that this construction is not a fully explicit expander is that it is not easy to find a prime such that it is close enough to an arbitrary number n.

Remark This graph can be constructed by Cayley graph with a non-abelian group with 3 generators. We have seen in problem set 3 that abelian group cannot give constant-degree expanders. This example shows that non-abelian group can.

Definition 6. *We call a graph G (n, d, γ)-graph if G has n vertices, degree d, and spectral expansion $\geq \gamma$.*

We will define three different operations and show their effects on the three parameters that we are interested in. Overall, we want to increase the number of nodes ($\Uparrow$), maintain the degree ($\sim$), and maintain the expansion ($\sim$).

2.1 Expander Operations: Squaring

Definition 7 (Squaring)**.** *If $G = (V, E)$ is a D-regular digraph, then $G^2 = (V, E')$ is a D^2-regular digraph on the same vertex set, where the (i, j)-th neighbor of a vertex x is the j-th neighbor of the i-th neighbor of vertex x in G. In particular, a random step on G^2 consists of two random steps on G.*

Intuitively, squaring is like taking two steps of random walk on G.

Theorem 8 (Squaring Operation)**.** *Given a $(n, d, 1-\omega)$-graph G, then G^2 is a $(n(\sim), d^2(\Uparrow), 1-\omega^2(\Uparrow))$-graph. In particular, if G is a (n, d, γ)-graph, then G^2 is a $(n, d^2, 2\gamma - \gamma^2 \approx 2\gamma)$-graph.*

Remark This theorem tells us that squaring maintains the number of nodes, increases the degree (worse), increases the expansion (better).

2.2 Expander Operations: Tensoring

Definition 9 (Tensoring)**.** *Let $G_1 = (V_1, E_1)$ be a D_1-regular graph and $G_2 = (V_2, E_2)$ be a D_2-regular graph. Then their tensor product is the $D_1 D_2$-regular graph $G_1 \otimes G_2 = (V_1 \times V_2, E)$, where the (i_1, i_2)-th neighbor of a vertex (x_1, x_2) is (y_1, y_2), where y_b is the i_b-th neighbor of vertex x_b in G_b.*

A random step in $G = G_1 \otimes G_2$ corresponds to a random step in G_1 in the first component and a random step in G_2 in the second component of $x = (x_1, x_2) \in G$.

Theorem 10 (Tensoring Operation)**.** *Given a (n, d, γ)-graph G, then $G \otimes G$ is a $(n^2(\Uparrow), d^2(\Uparrow), \gamma(\sim))$-graph.*

Remark This theorem tells us that tensoring increases the number of nodes (good), increases the degree (bad), maintains the expansion (same).

Here are some general properties of tensor product of two graphs:

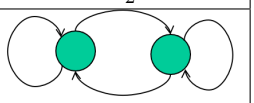
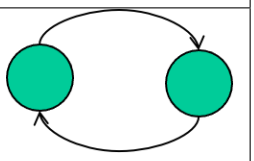
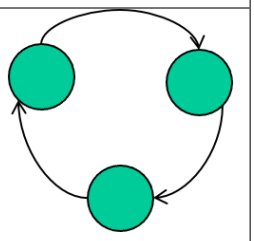
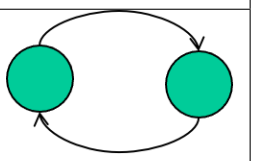
Theorem 11 (Properties of Tensoring)**.** *Given a (n_1, d_1, γ_1)-graph G_1 and a (n_2, d_2, γ_2)-graph G_2, we have:*

- *The vertex set of the tensor product $G = G_1 \otimes G_2$ is $V = V_1 \times V_2$*
- *Edge weight of G is $w_{(a_1,a_2),(b_1,b_2)} = w_{(a_1,b_1)} w_{(a_2,b_2)}$.*

- *Adjacency matrix* $M = M_1 \otimes M_2$.
- *Eigenvector of* W: $\{v_1 \otimes v_2 \mid v_1$ *eigenvector of* W_1, v_2 *eigenvector of* $W_2\}$, *where* $(v \otimes w)(a,b) = v(a)w(b)$.
- *Eigenvalues of* W: $\{\omega_1\omega_2 \mid \omega_1$ *eigenvalue of* W_1, ω_2 *eigenvalue of* $W_2\}$.
- *Spectral expansion:* $\min(\gamma(G_1), \gamma(G_2)) = 1 - \max(\omega(W_1), \omega(W_2))$.

Remark Although we have eigenvalues being the product of the original eigenvalues, only the largest one 1×1 is preserved, while others are of form $\omega_1\omega_2$. Thus the second largest eigenvalue is the maximum of $1 \times \omega(W_1)$ and $1 \times \omega(W_2)$.

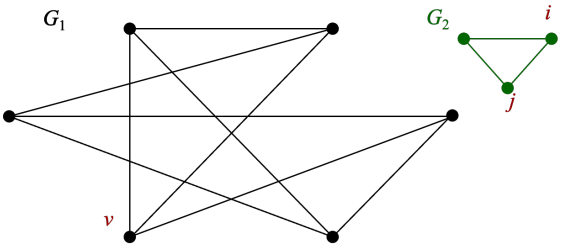
Example 12. *In this example, we show the tensor product of* M_1 *and some common matrices.*

$M = M_1 \otimes M_2$	G_2
$\begin{pmatrix} M_1 & M_1 \\ M_1 & M_1 \end{pmatrix} = M_1 \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	
$\begin{pmatrix} 0 & M_1 \\ M_1 & 0 \end{pmatrix} = M_1 \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
$\begin{pmatrix} 0 & 0 & M_1 \\ M_1 & 0 & 0 \\ 0 & M_1 & 0 \end{pmatrix} = M_1 \otimes \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	
$\begin{pmatrix} M_1(1,1) & 0 & \dots & M_1(1,n) & 0 \\ 0 & M_1(1,1) & \dots & 0 & M_1(1,n) \\ & & \dots & & \\ M_1(n,1) & 0 & \dots & M_1(n,n) & 0 \\ 0 & M_1(n,1) & \dots & 0 & M_1(n,n) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes M_1$	

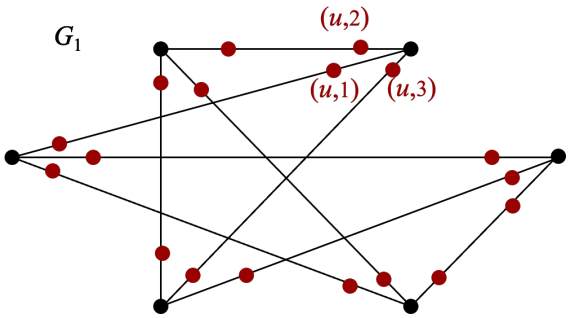
2.3 Expander Operations: Zig-Zag Product

Definition 13 (Zig-Zag Product)**.** *Let* G_1 *be a* D_1*-regular graph on* N_1 *vertices, and* G_2 *a* D_2*-regular graph on* D_1 *vertices. Then* $G_1 \textcircled{z} G_2$ *is a graph whose vertices are pairs* $(u,i) \in [N_1] \times [D_1]$ *with degree* D_2^2*. For* $a, b \in [D_2]$*, the* (a,b)*-th neighbor of a vertex* (u,i) *is the vertex* (v,j) *defined as follows:*

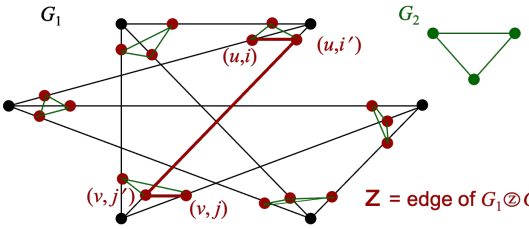
1. *Let* i' *be the* a*-th neighbor of vertex* i *in* G_2.
2. *Let* v *be the* i'*-th neighbor of* u *in* G_1. *Let* j' *be the index such that edge* $e = (u,v)$ *is the* j'*-th edge entering* v *in* G_1.
3. *Let* j *be the* b*-th neighbor of vertex* j' *in* G_2.



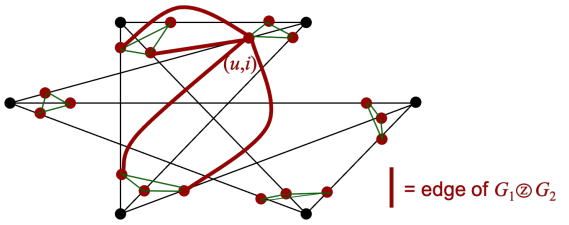
(a) Initial D_1-regular graph G_1 and D_2-regular G_2 with size D_1.



(b) First we construct a cloud with the same shape as G_2 for each vertex $u \in G_1$.



(c) Three steps connecting (u, i) and (v, j) following Definition 13



(d) We construct an edge between (u, i) and (v, j) and similarly for other edges.

Figure 1: Visualization of the edge construction in the Zig-Zag product $G_1 ⓩ G_2$.

Remark Zig-Zag product is very similar to tensor product of two graphs. But instead of having the random walks on two graphs move independently, Zig-Zag product lets the second graph determines the random walk of the first graph. In the second graph, we still do a purely random walk on it.

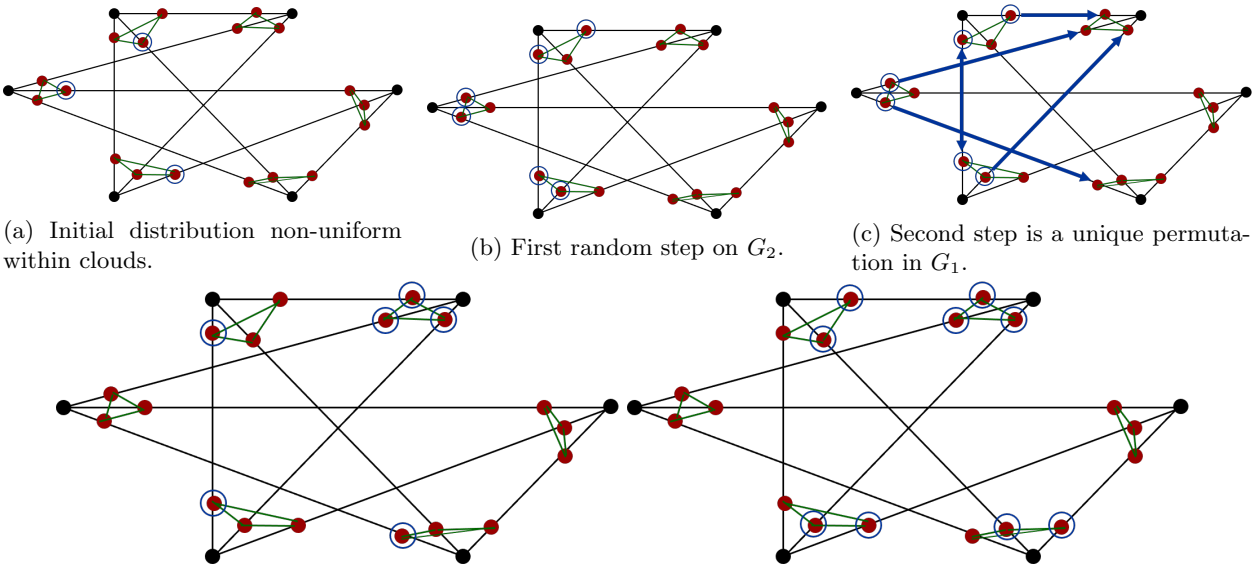
2.3.1 Why Zig-Zag Product Can Improve the Expansion?

We want to intuitively check whether the three steps in the Zig-Zag product can move a non-uniform distribution toward a uniform distribution. We split it into two cases:

Case 1: when the distribution is not uniform inside the clouds In this case, we assume the initial distribution is not uniform in clouds, i.e., some vertices in a cloud are contained in the distribution but some are not. Illustration is shown in Figure 3.

In the first random step in G_2, we know that the Zig-Zag product can make a non-uniform distribution within clouds move a random step in clouds and therefore toward a more uniform distribution. In the second step, the step follows a unique permutation defined by the labeling of G_1 and G_2. Since permutation does not affect the entropy, the overall randomness or uniformness is not affected by the second step. Lastly, the third step is another random walk in G_2, which can only increase the randomness. Therefore, intuitively we conclude that the Zig-Zag product can move toward a more uniform distribution.

Case 2: when the distribution is uniform inside the clouds In case 2, we assume the distribution is uniform within clouds but not uniform across clouds. In the first random step in G_2, the distribution does not get any closer to the uniform distribution because it has already been uniform within clouds. But in the second step, the unique permutation in G_1 can distribute the vertices in clouds to other clouds. This gives us an opportunity that we can spread the vertices in the third random step in G_2 because it is no longer a uniform distribution within clouds.



(a) Initial distribution non-uniform within clouds.

(b) First random step on G_2.

(c) Second step is a unique permutation in G_1.

(d) After taking the second step.

(e) Third step is a random step on G_2.

Figure 2: Illustration of why in Case 1 the Zig-Zag product can bring a non-uniform distribution in clouds toward a more uniform distribution in clouds.

Remark If we only have the first two steps of the Zig-Zag product, then its squaring (two-step random walk) still mixes quickly. However, as we have seen in the problem set 1, two-step random walk is not completely equivalent to one-step random walk. This is one reason why we want to define the Zig-Zag product by three symmetric steps. Another reason as we have seen in the reading is that we want the graph to be undirected. But in fact there is nothing preventing us from using a directed graph.

2.3.2 Linear Algebra Analysis of Zig-Zag Product

Theorem 14 (Zig-Zag product)**.** *Let G_1 be a (n_1, d_1, γ_1)-graph and G_2 be a (n_2, d_2, γ_2)-graph. The Zig-Zag product $G_1 \textcircled{z} G_2$ is a $(n_1 d_1 (\Uparrow), d_2^2 (\Downarrow), \gamma_1 \gamma_2^2 (\Downarrow))$.*

Or we can simply write:

$$(n_1, d_1, \gamma_1) \textcircled{z} (d_1, d_2, \gamma_2) \to (n_1 d_1, d_2^2, \gamma_1 \gamma_2^2) \tag{6}$$

and in terms of ω:

$$(n_1, d_1, 1-\omega_1) \textcircled{z} (d_1, d_2, 1-\omega_2) \to (n_1 d_1, d_2^2, 1-\omega_1-2\omega_2) \tag{7}$$

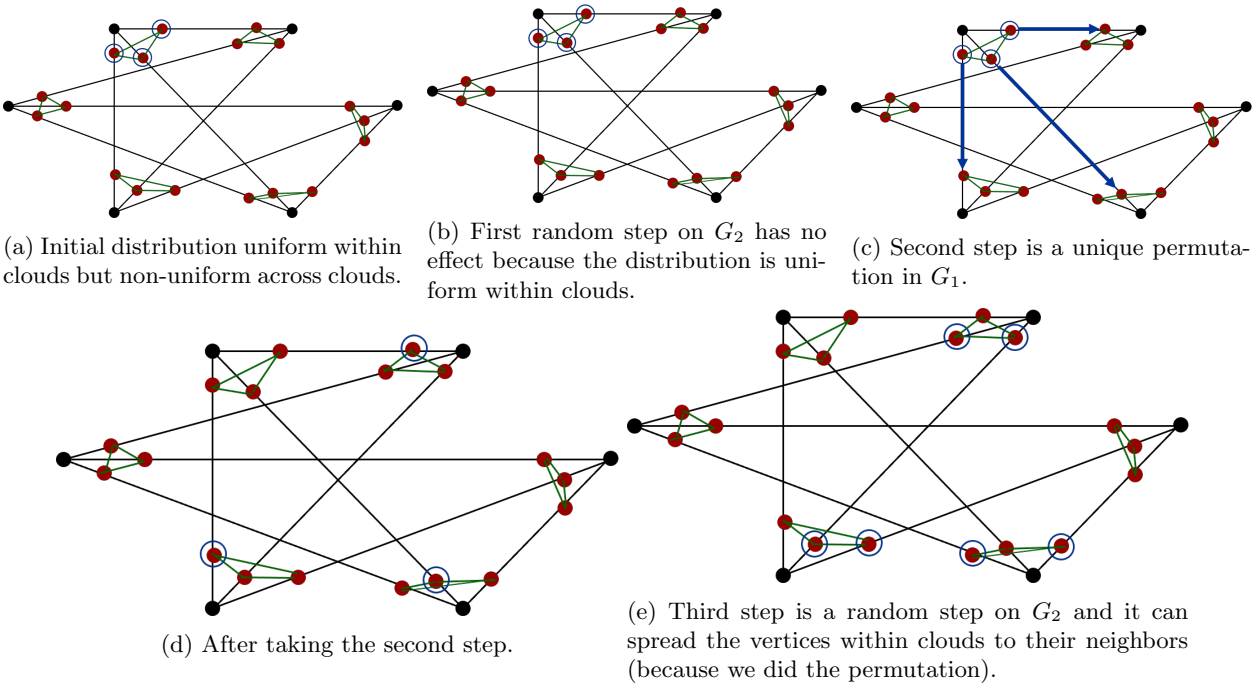
when ω_1, ω_2 are small.

Proof. As we have seen before, the three steps in the Zig-Zag product can be intepreted as i) a random walk on G_2 ii) a permutation on G_1 iii) a random walk on G_2. Thus we can write the Zig-Zag product as:

$$W_{\text{zigzag}} = \tilde{W}_2 \hat{W}_1 \tilde{W}_2 \tag{8}$$

where $\hat{W}_1$ is a permutation matrix; $\hat{W}_1((b,j),(a,i)) = \begin{cases} 1 \text{ if } i\text{-th edge leaving } a = j\text{-th edge leaving } b \\ 0 \text{ otherwise} \end{cases}$

We also know that the random walk matrix on G_2 is given by: $\tilde{W}_2 = I_{n_1} \otimes W_2$ the tensor product of an identity matrix with size n_1 the number of vertices in G_1, and the random walk matrix W_2 of G_2. The



(a) Initial distribution uniform within clouds but non-uniform across clouds.

(b) First random step on G_2 has no effect because the distribution is uniform within clouds.

(c) Second step is a unique permutation in G_1.

(d) After taking the second step.

(e) Third step is a random step on G_2 and it can spread the vertices within clouds to their neighbors (because we did the permutation).

Figure 3: Illustration of why in Case 2 the Zig-Zag product can bring a non-uniform distribution across clouds toward a more uniform distribution across clouds.

reason that this is true is because we simultaneously run random walk on all the clouds. So it is identical to having a tensor product of a graph with no edge in between clouds (identity matrix) and the random walk matrix of G_2. Therefore, we can write the matrix $\tilde{W}_2$ by:

$$\tilde{W}_2 = I_{n_1} \otimes W_2 = I_{n_1} \otimes (\gamma_2 J_{n_2} + (1-\gamma_2)E_2) = \gamma_2 \tilde{J}_{n_2} + (1-\gamma_2) I_{n_1} \otimes E_2 \tag{9}$$

where in the later part we apply the matrix decomposition of W_2 and $\tilde{J}_{n_2} = I_{n_1} \otimes J_{n_2}$ is the random walk matrix of a complete graph with size n_2 but with n_1 repetition to each node as we have shown in Example 12. But it can be interpreted as n_1 copies of a complete graph with size n_2.

Plugging the above expression of $\tilde{W}_2$ into Equation 8, we can get:

$$\begin{aligned} W_{\text{zigzag}} &= \tilde{W}_2 \hat{W}_1 \tilde{W}_2 \\ &= (\gamma_2 \tilde{J}_{n_2} + (1-\gamma_2) I_{n_1} \otimes E_2) \hat{W}_1 (\gamma_2 \tilde{J}_{n_2} + (1-\gamma_2) I_{n_1} \otimes E_2) \\ &= \gamma_2^2 \tilde{J}_{n_2} \tilde{W}_1 \tilde{J}_{n_2} + (1-\gamma_2^2) F \end{aligned} \tag{10}$$

where F is the sum of the other terms involving E_2 and we know that $\|F\| \leq 1$.

Observation 15. *If we apply Zig-Zag product to G_1 and a complete graph K_{n_2}, it is exactly the same as a tensor product.*

This observation tells us that $G_1 \textcircled{z} K_{n_2} = G_1 \otimes K_{n_2}$. What we have in hand is a graph composed of n_1 copies of K_{n_2}, where its Zig-Zag product is still the same as the tensor product. By this observation, we can therefore compute the first term in Equation 10 by:

$$\tilde{J}_{n_2} \tilde{W}_1 \tilde{J}_{n_2} = \tilde{W}_1 \textcircled{z} \tilde{J}_{n_2} = \tilde{W}_1 \otimes \tilde{J}_{n_2} \tag{11}$$

Therefore we can finally bound Equation 10:

$$W_{\text{zigzag}} = \gamma_2^2 \tilde{J}_{n_2} \hat{W}_1 \tilde{J}_{n_2} + (1-\gamma_2^2) F = \gamma_2^2 \hat{W}_1 \otimes \tilde{J}_{n_2} + (1-\gamma_2^2) F \tag{12}$$

Therefore we can also bound the second largest eigenvalue by:

$$\begin{aligned}\lambda(W_{\text{zigzag}}) &\leq \gamma_2^2 \max(\lambda(\tilde{J}_{n_2}), \lambda(\tilde{W}_1)) + (1-\gamma_2^2) \\ &\leq \gamma_2^2(1-\gamma_1) + (1-\gamma_2^2) \\ &= 1 - \gamma_1\gamma_2^2 \end{aligned} \tag{13}$$

where the first inequality comes from the properties of spectral expansion of tensor product. □

Remark Combining three expander operations and their expansion properties in Theorem 8, Theorem 10, and Theorem 14, we can balance the degree and the spectral expansion so that they remain constant, while increasing the number of vertices to construct a family of expander graphs.