On the trade-off between expressivity & privacy in graph representation learning

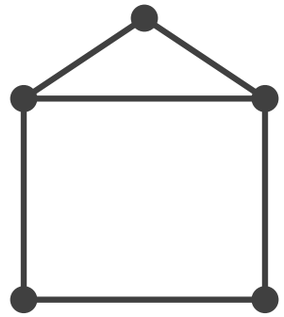
Patrick Indri*, Tamara Drucks*, Thomas Gärtner

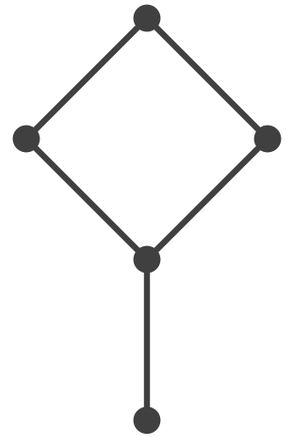
TECHNISCHE
UNIVERSITÄT
WIEN

ICLR

What's the deal

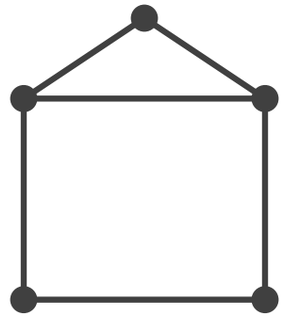
In graph ML we want *good* representations $\varphi(G)$ of a graph G

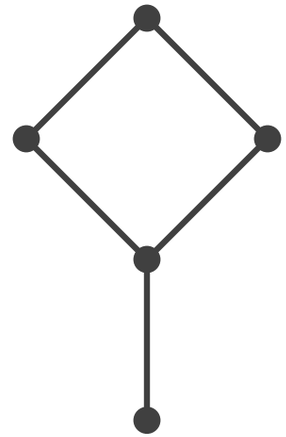
G_1 

G_2 

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

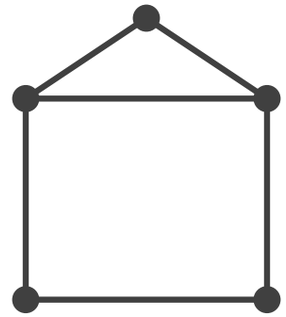
G_1 

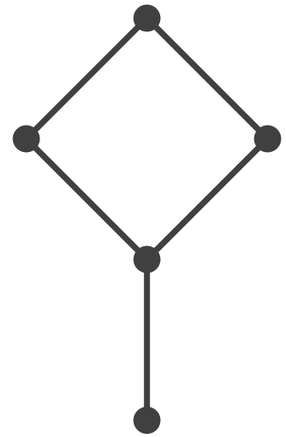
G_2 

But what's *good*? Depends on who you ask

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

G_1 

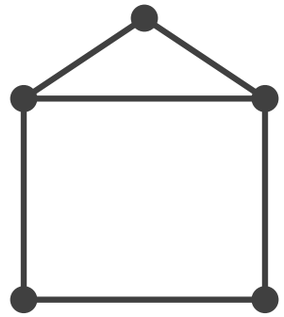
G_2 

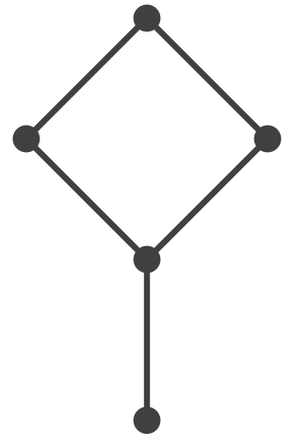
But what's *good*? Depends on who you ask

Let's ask an **expressivity** person!

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

G_1 

G_2 

But what's *good*? Depends on who you ask

Let's ask an **expressivity** person!

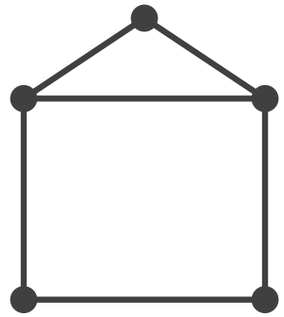
Tamara

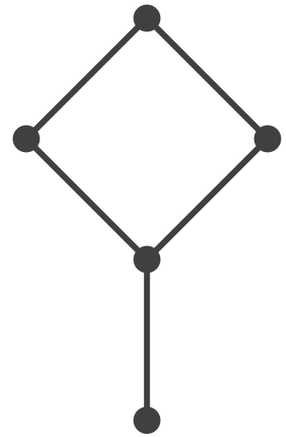
$$\varphi(G_1) = \varphi(G_2) \iff G_1 \simeq G_2$$

Same representations iff the graphs are isomorphic

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

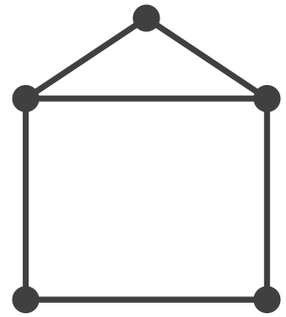
G_1 

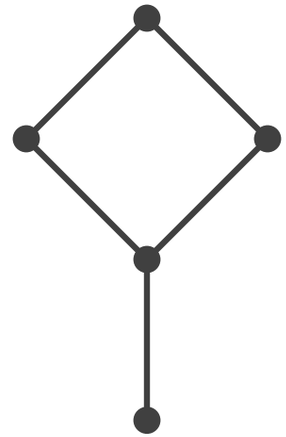
G_2 

But what's *good*? Depends on who you ask

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

G_1 

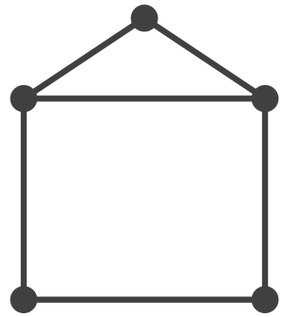
G_2 

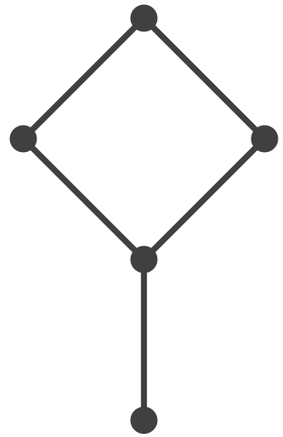
But what's *good*? Depends on who you ask

Let's ask a **privacy** person!

What's the deal

In graph ML we want *good* representations $\varphi(G)$ of a graph G

G_1 

G_2 

But what's *good*? Depends on who you ask

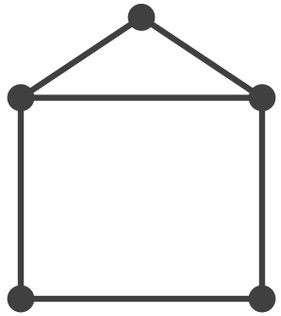
Let's ask a **privacy** person!

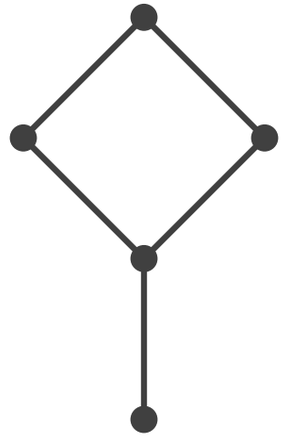
Patrick

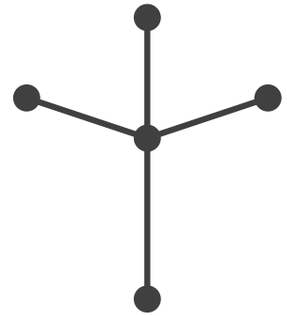
$$d_{\text{edge}}(G_1, G_2) = 1 \implies \varphi(G_1) \approx \varphi(G_2)$$

Similar representations for edge-adjacent graphs

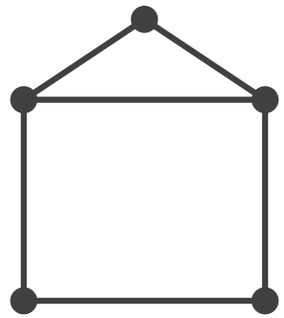
Why not both?

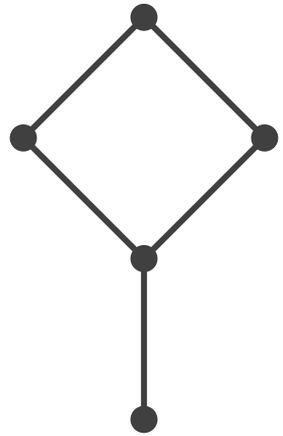
G_1 

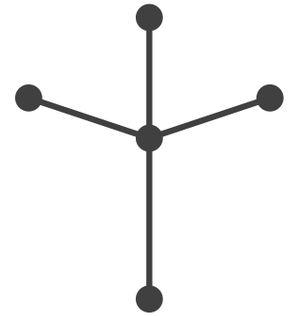
G_2 

G_3 

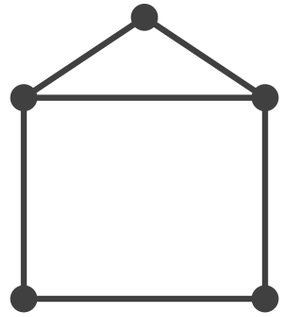
Why not both?

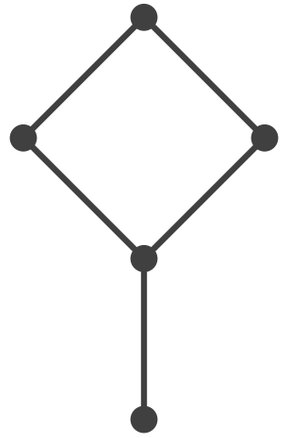
G_1 

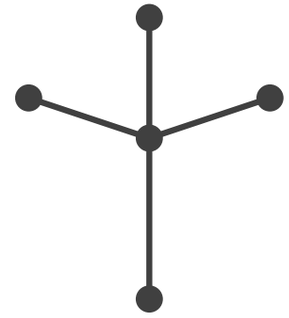
G_2 

G_3 

Why not both?

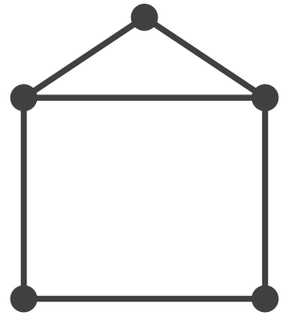
G_1 

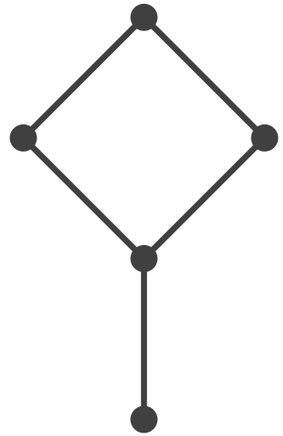
G_2 

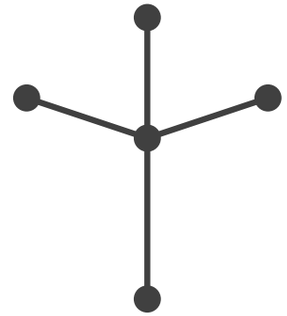
G_3 

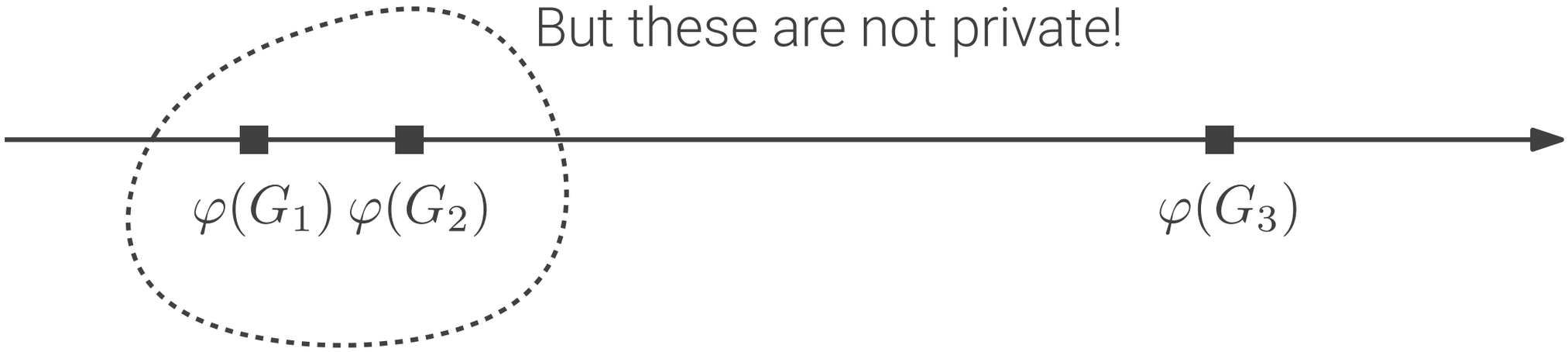
The **expressivity** person wants: $\varphi(G_1) = \varphi(G_2) \iff G_1 \simeq G_2$

Why not both?

G_1 

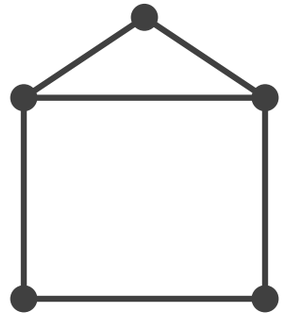
G_2 

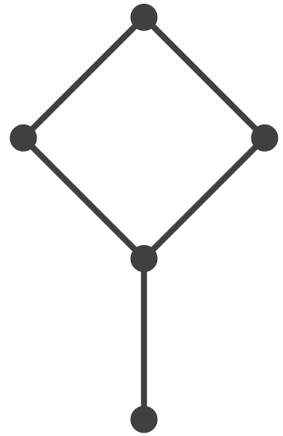
G_3 

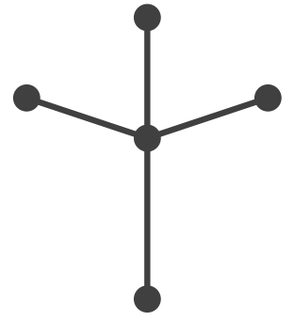


The **expressivity** person wants: $\varphi(G_1) = \varphi(G_2) \iff G_1 \simeq G_2$

Why not both?

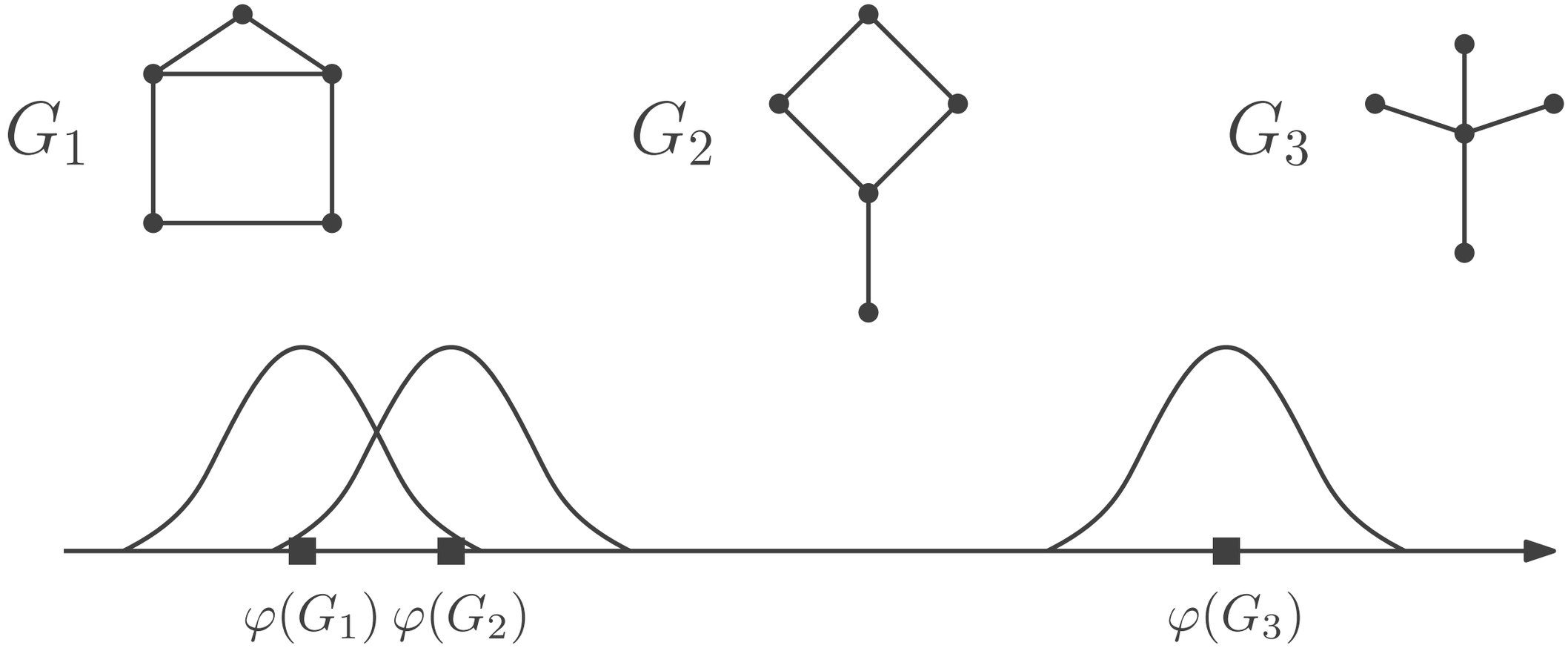
G_1 

G_2 

G_3 

The **privacy** person wants: $d_{\text{edge}}(G_1, G_2) = 1 \implies \varphi(G_1) \approx \varphi(G_2)$

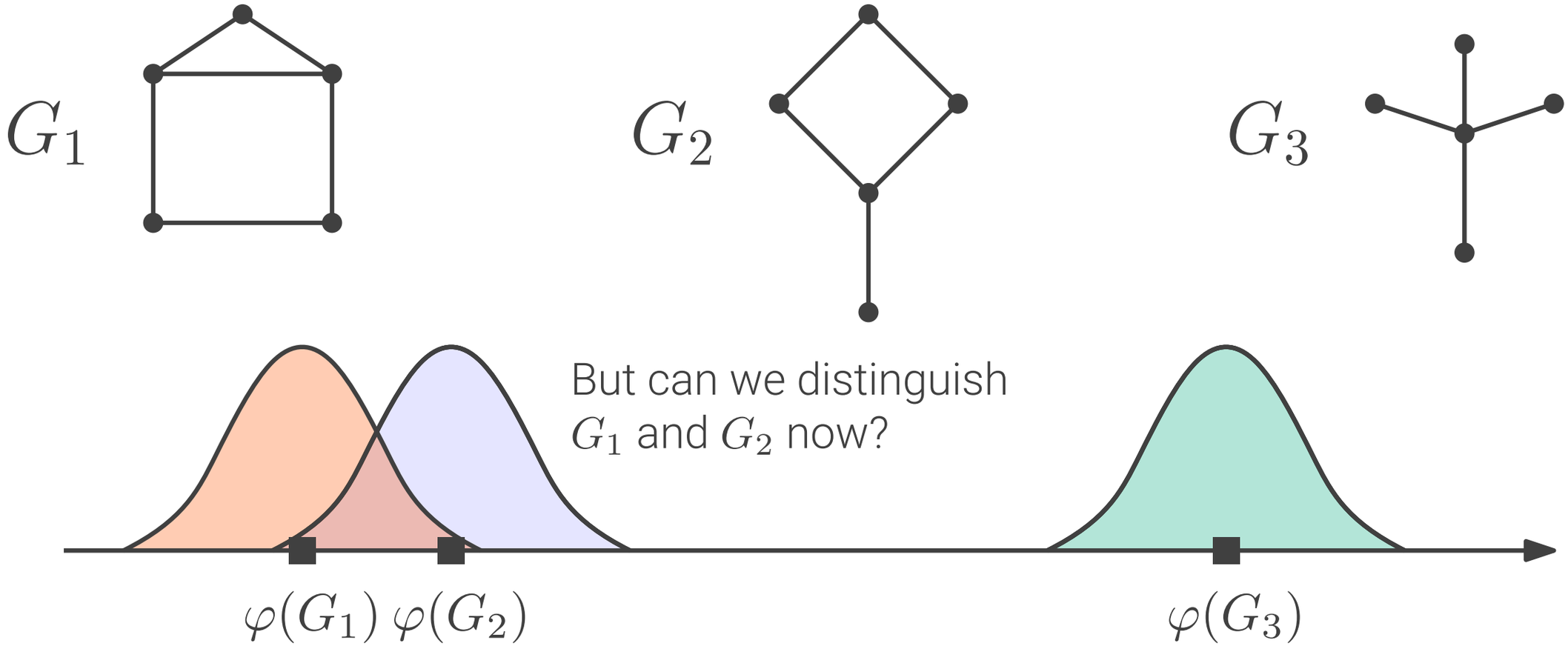
Why not both?



let's add noise to the embeddings!

The **privacy** person wants: $d_{\text{edge}}(G_1, G_2) = 1 \implies \varphi(G_1) \approx \varphi(G_2)$

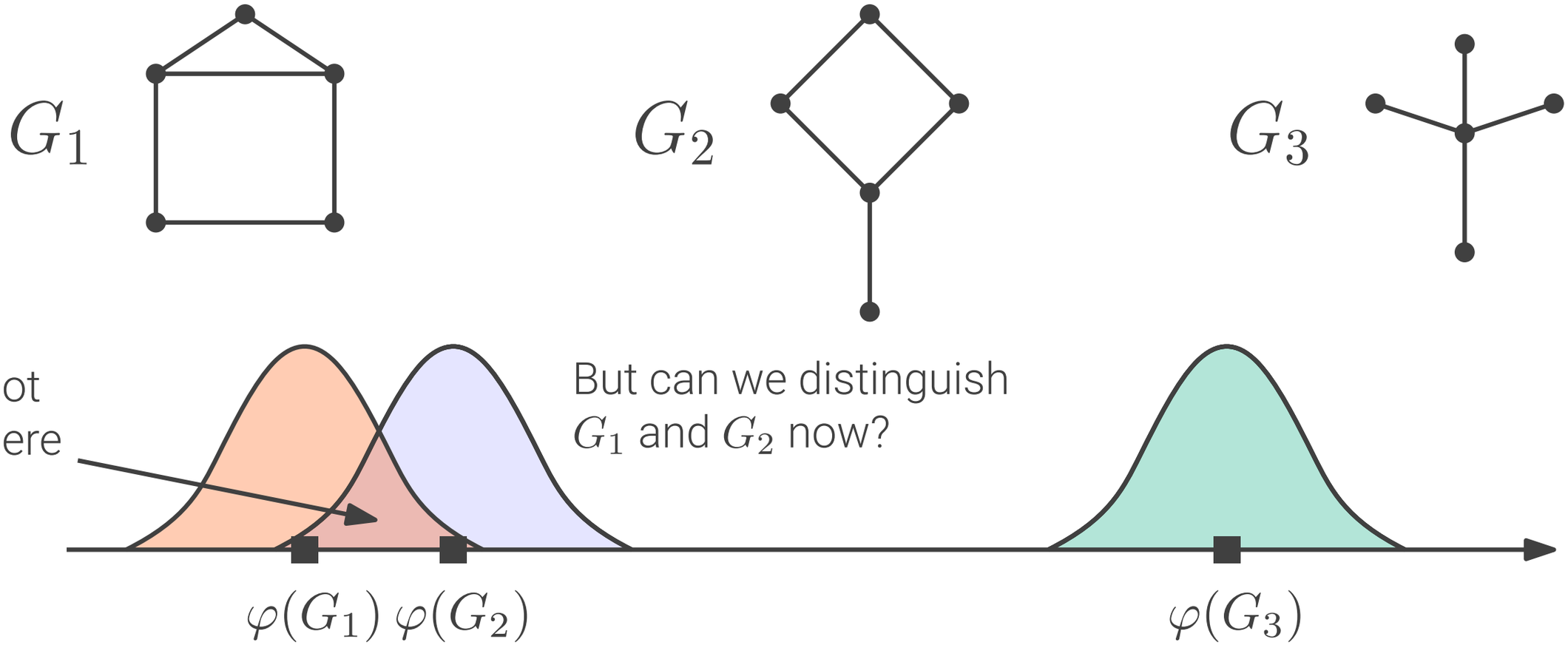
Why not both?



let's add noise to the embeddings!

The **privacy** person wants: $d_{\text{edge}}(G_1, G_2) = 1 \implies \varphi(G_1) \approx \varphi(G_2)$

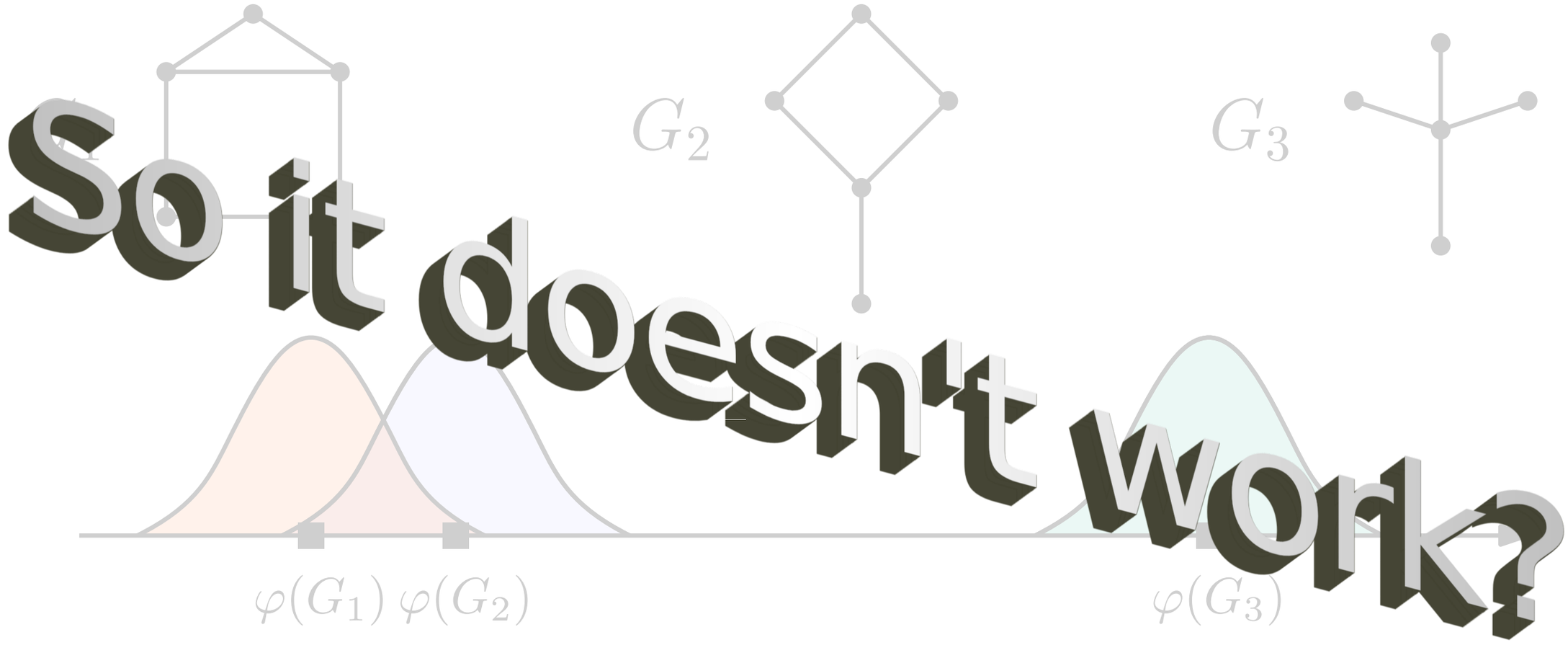
Why not both?



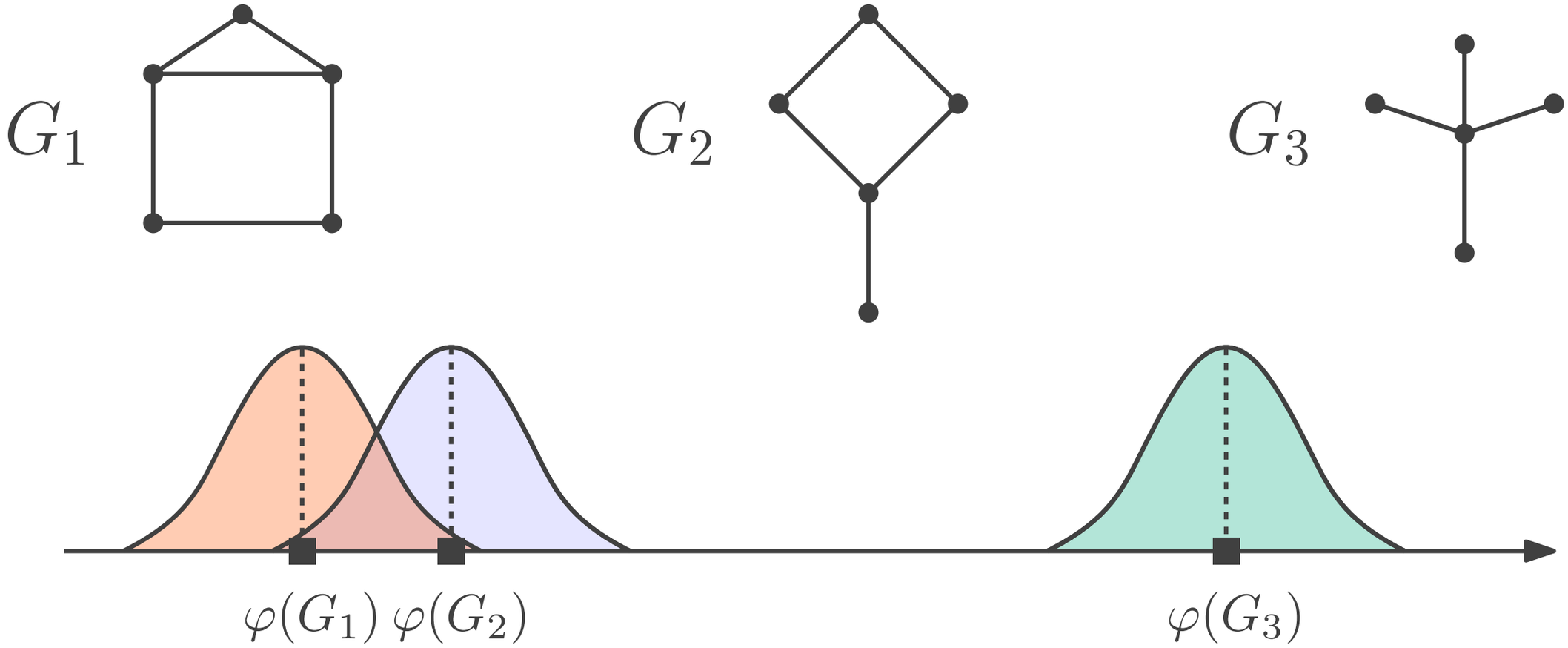
let's add noise to the embeddings!

The **privacy** person wants: $d_{\text{edge}}(G_1, G_2) = 1 \implies \varphi(G_1) \approx \varphi(G_2)$

Why not both?

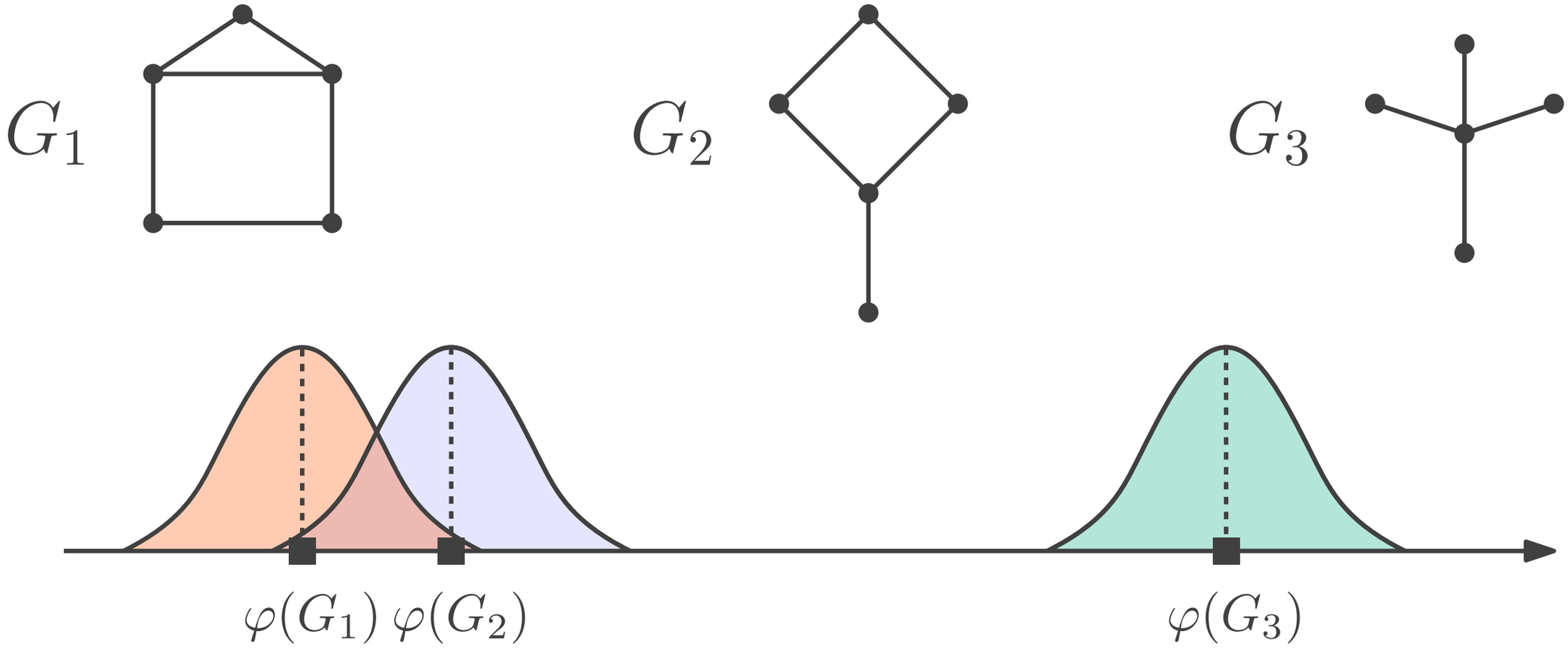


We can make both work! Sort of



The **expected value** is still what the expressivity person wants

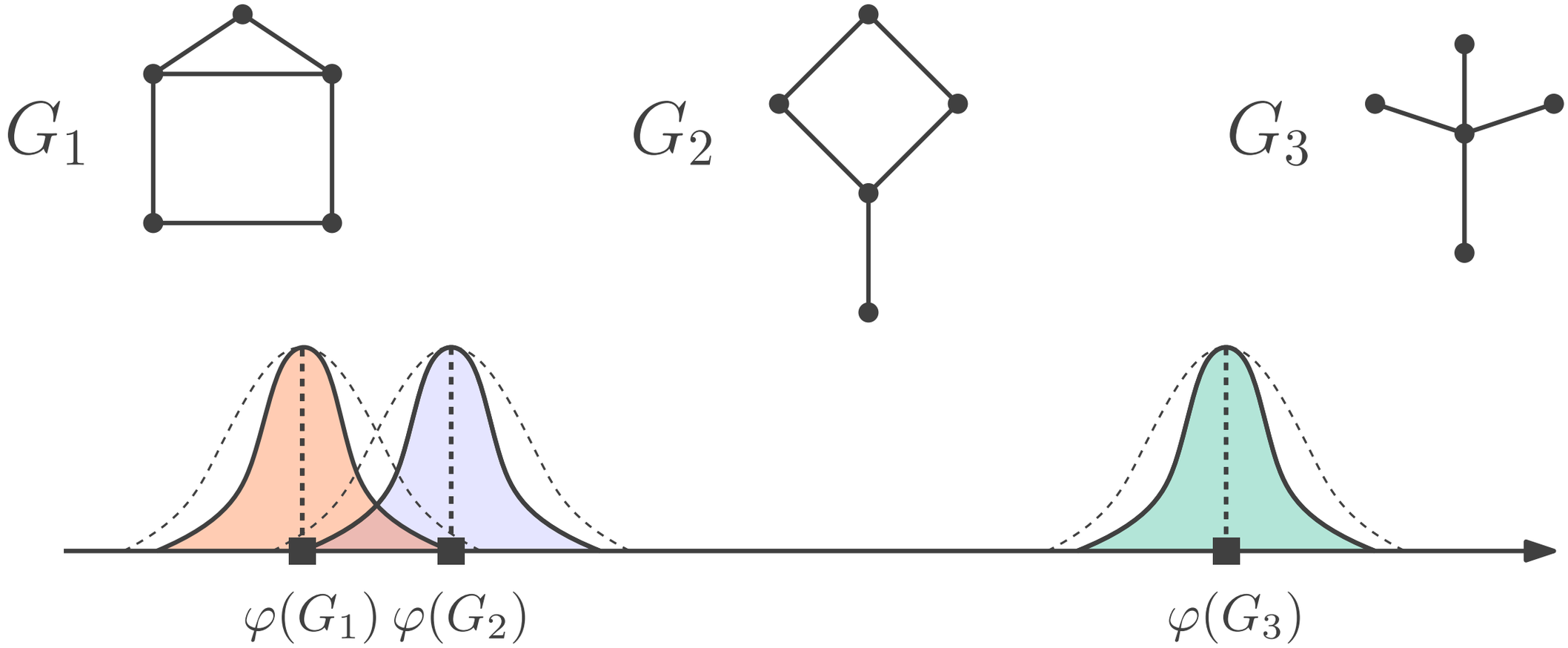
We can make both work! Sort of



The **expected value** is still what the expressivity person wants

And with a good **noise calibration** we can still be private

We can make both work! Sort of



The **expected value** is still what the expressivity person wants

And with a good **noise calibration** we can still be private

What we do

We consider expressivity **in expectation** and a **local** notion of differential privacy.

What we do

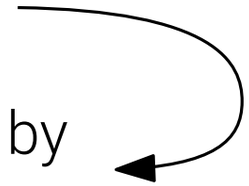
We consider expressivity **in expectation** and a **local** notion of differential privacy.

Result 1. We obtain graph embeddings that are **expressive** in expectation and differentially **private**.

What we do

We consider expressivity **in expectation** and a **local** notion of differential privacy.

Result 1. We obtain graph embeddings that are **expressive** in expectation and differentially **private**.



What we do

We consider expressivity **in expectation** and a **local** notion of differential privacy.

Result 1. We obtain graph embeddings that are **expressive** in expectation and differentially **private**.

determined by *pattern class*

Result 2. **More expressive** pattern classes often require **more noise** to be added to guarantee privacy.

How?

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Pattern graphs $(F_1, \ldots, F_n)$

$\varphi(G) =$

F_1	F_2	F_3		F_n
3	27	12		9

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

Pattern graphs $(F_1, \ldots, F_n)$

$\varphi(G) =$

F_1	F_2	F_3		F_n
3	27	12		9

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

number of homomorphisms from F to G'

$$G \simeq G' \iff \hom(F, G) = \overbrace{\hom(F, G')}$$ **for all** simple graphs F

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi(G) =$$

F_1	F_2	F_3		F_n
3	27	12		9

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

number of homomorphisms from F to G'

$$G \simeq G' \iff \hom(F, G) = \overbrace{\hom(F, G')} \text{ **for all** simple graphs } F$$

Homomorphism counts give us a **complete** embedding

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi(G) = \begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & \cdots & F_n \\ \hline 3 & 27 & 12 & & 9 \\ \hline \end{array}$$

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

number of homomorphisms from F to G'

$$G \simeq G' \iff \hom(F, G) = \hom(F, G') \text{ **for all** simple graphs } F$$

Homomorphism counts give us a **complete** embedding

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi(G) = \begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & & F_n \\ \hline 3 & 27 & 12 & & 9 \\ \hline \end{array}$$

complete in expectation

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

$G \simeq_{\mathcal{F}} G' \iff \hom(F, G) = \hom(F, G')$ **for all** $F \in \mathcal{F}$ ← restrict pattern class

Pattern graphs $(F_1, \ldots, F_n)$

$\varphi(G) =$

F_1	F_2	F_3		F_n
3	27	12		9

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

$G \simeq_{\mathcal{F}} G' \iff \hom(F, G) = \hom(F, G')$ **for all** $F \in \mathcal{F}$ ← restrict pattern class

Homomorphism counts give us an **expressive** embedding

Pattern graphs $(F_1, \ldots, F_n)$

$\varphi(G) =$

F_1	F_2	F_3		F_n
3	27	12		9

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

Why homomorphisms?

$G \simeq_{\mathcal{F}} G' \iff \hom(F, G) = \hom(F, G')$ **for all** $F \in \mathcal{F}$ ← restrict pattern class

Homomorphism counts give us an **expressive** embedding

Pattern graphs $(F_1, \ldots, F_n)$

$\varphi(G) =$

F_1	F_2	F_3		F_n
3	27	12		9

expressive in expectation

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

We add noise to the normalized counts to be **differentially private**

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi_{\mathrm{DP}}(G) = \begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & & F_n \\ \hline 0.2 & 0.8 & 0.3 & & 0.1 \\ \hline \end{array}$$

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

We add noise to the normalized counts to be **differentially private**

Why differential privacy?

$\varphi_{\mathrm{DP}}(G) \approx \varphi_{\mathrm{DP}}(G')$ **for all** G, G' with $d_{\mathsf{edge}}(G, G') = 1$

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi_{\mathrm{DP}}(G) = \begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & \cdots & F_n \\ \hline 0.2 & 0.8 & 0.3 & & 0.1 \\ \hline \end{array}$$

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

We add noise to the normalized counts to be **differentially private**

Why differential privacy?

$\varphi_{\mathrm{DP}}(G) \approx \varphi_{\mathrm{DP}}(G')$ **for all** G, G' with $d_{\mathsf{edge}}(G, G') = 1$

How much noise?

We add **small-ish** noise that depends on a **local** notion of privacy

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi_{\mathrm{DP}}(G) = \begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & \cdots & F_n \\ \hline 0.2 & 0.8 & 0.3 & & 0.1 \\ \hline \end{array}$$

How?

We count **homomorphisms** from **sampled** pattern graphs F_i to G

We add noise to the normalized counts to be **differentially private**

Pattern graphs $(F_1, \ldots, F_n)$

$$\varphi_{\mathrm{DP}}(G) = \underbrace{\begin{array}{|c|c|c|c|c|} \hline F_1 & F_2 & F_3 & & F_n \\ \hline 0.2 & 0.8 & 0.3 & & 0.1 \\ \hline \end{array}}$$

expressive in expectation

To wrap up

TLDR: We sample **noisy homomorphism densities** to obtain **provably** expressive and private graph representations

By kinda relaxing privacy a bit and looking at a *local* notion of it

By kinda relaxing expressivity and ensuring it in *expectation*

To wrap up

TLDR: We sample **noisy homomorphism densities** to obtain **provably** expressive and private graph representations

By kinda relaxing privacy a bit and looking at a *local* notion of it

By kinda relaxing expressivity and ensuring it in *expectation*

We also run **experiments** and they work!

To wrap up

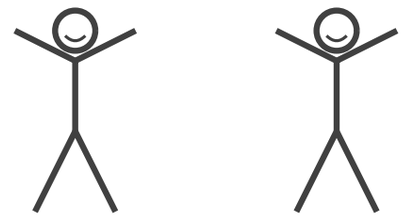
TLDR: We sample **noisy homomorphism densities** to obtain **provably** expressive and private graph representations

By kinda relaxing privacy a bit and looking at a *local* notion of it

By kinda relaxing expressivity and ensuring it in *expectation*

We also run **experiments** and they work!

THANKS:)



Tamara Patrick

Both moderately happy