

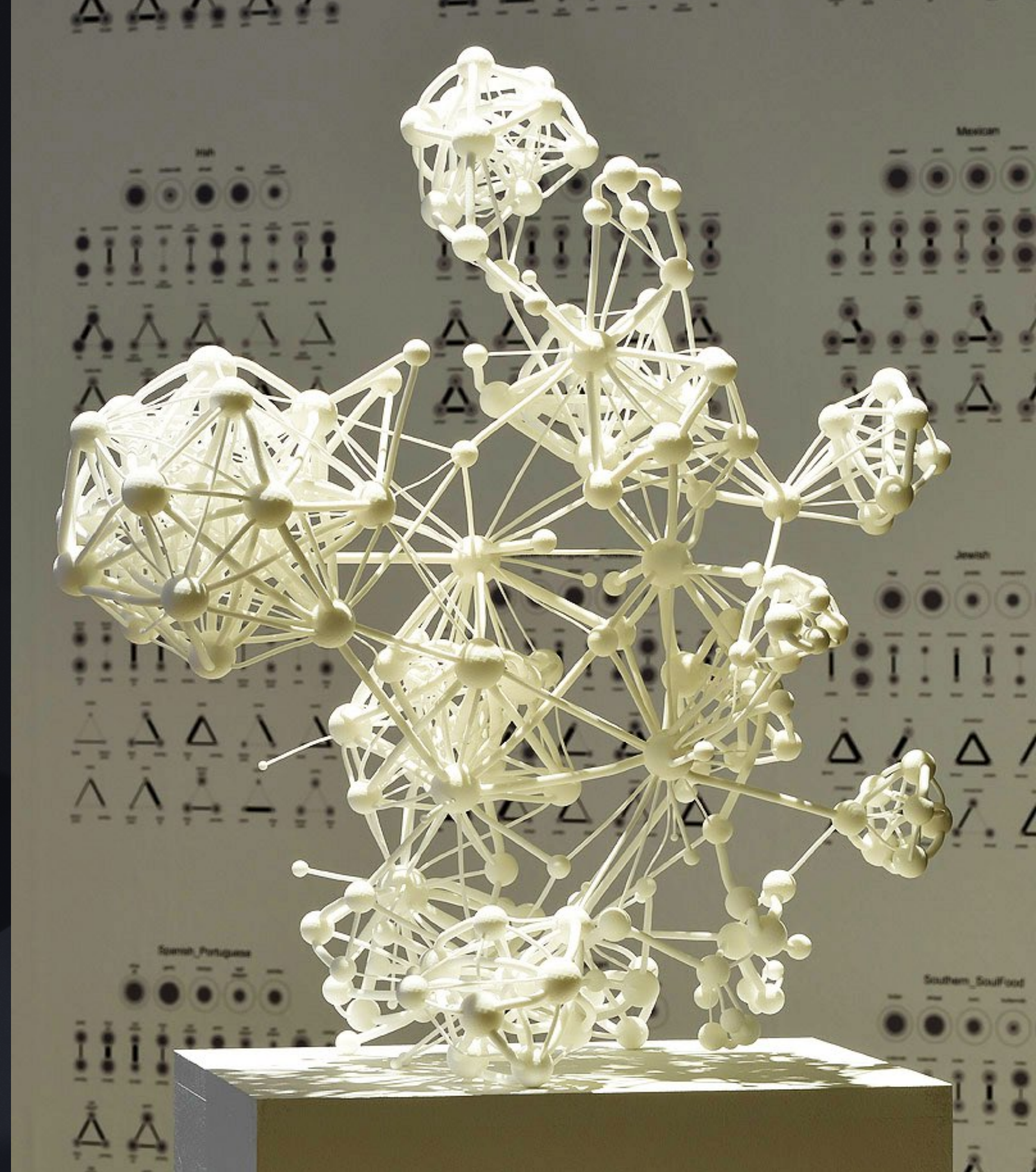
Your Lightning Network Simulation is Lying to You

The real shape of the Lightning Network and how to match it

The Problem

Wrong graph in, wrong answer out

- Pathfinding, routing, fees, attacks are all studied and simulated on a graph.
- Experimenting on mainnet is mostly unfeasible, so the graph is synthetic.
- **If the graph shape is wrong, the result is a property of that wrong graph.**

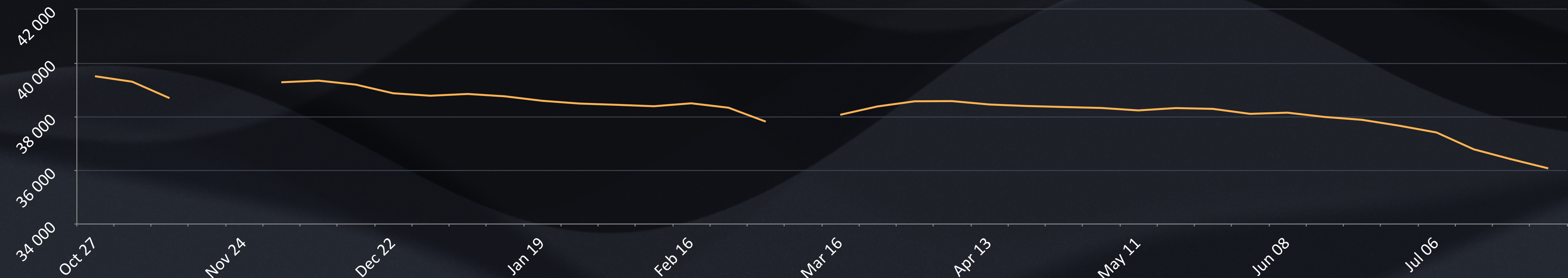


The Data

Nine months of hourly measurements



node count



channel count

The Target

The fingerprint a model must match

metric	Lightning Network
mean degree	6.33
share of degree-1 nodes	60%
share of degree-3-or-less nodes	80%
p99 degree	89
max degree	1794
transitivity (triangle density)	0.040
average local clustering	0.108
degree assortativity	-0.254
radius	5
diameter	9

The Target, Over Time

The fingerprint does not move



The Standard Models

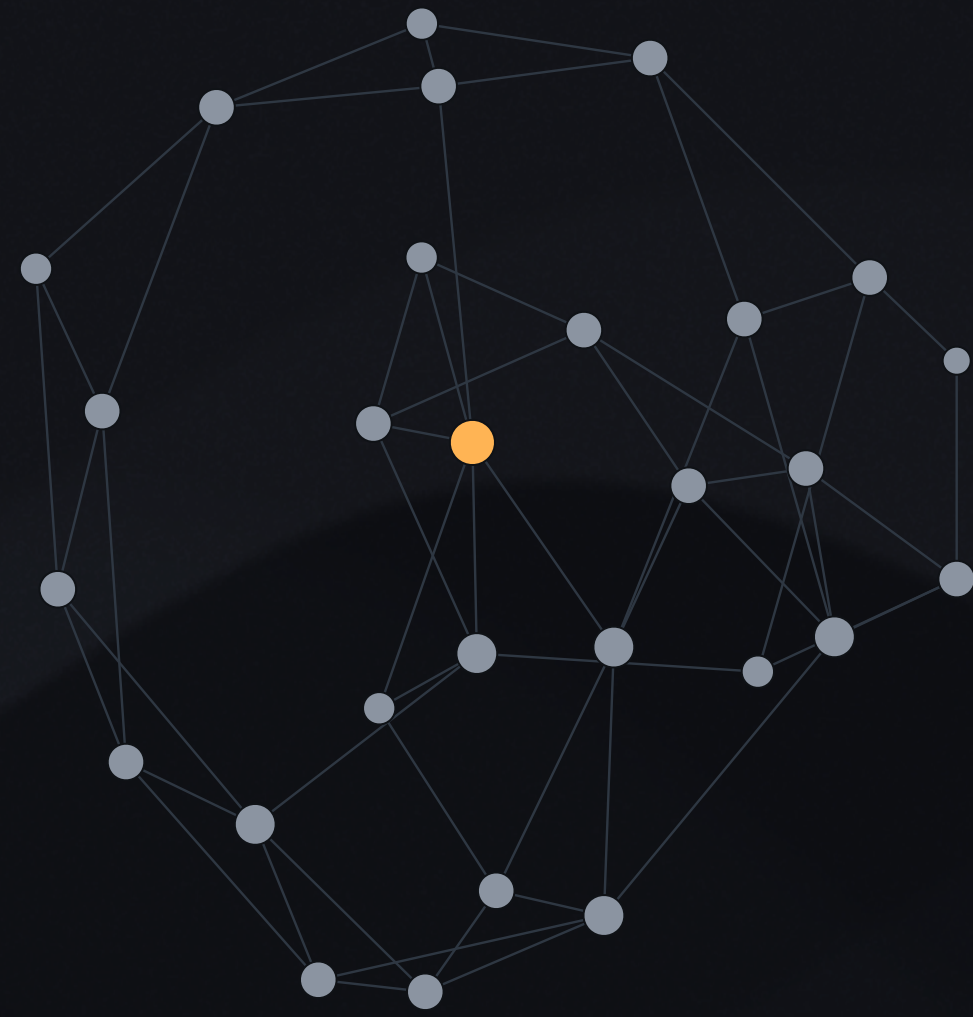
Four generators, one rule each

Erdős–Rényi



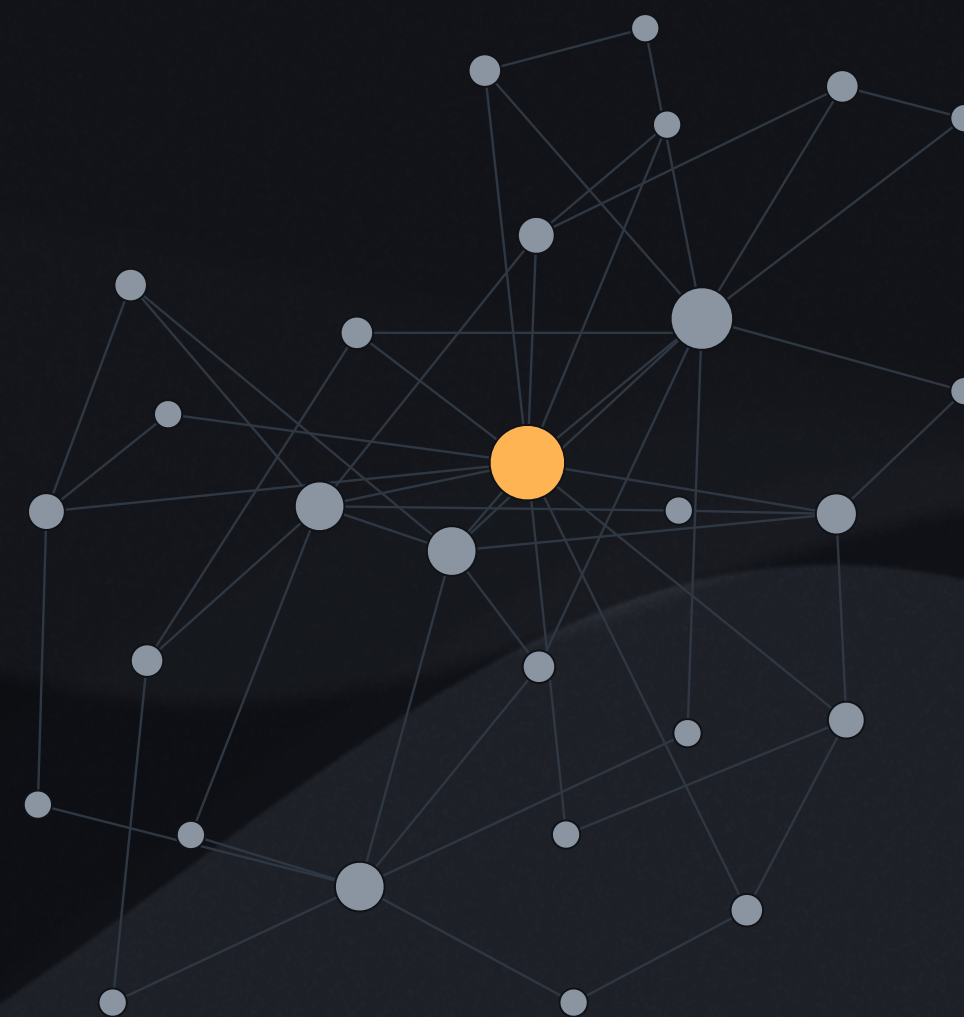
connect random pairs;
every pair equally likely

Watts–Strogatz



a ring of neighbors,
a few random shortcuts

Barabási–Albert



newcomers attach to
already-popular nodes

disassortative dynamic
Barabási–Albert



Barabási–Albert, where links
between leaves keep dying

White Paper at RECSI 2026

No standard model matches

does it reproduce Lightning Network... ?	ER	WS	BA	ddBA
heavy tail + big hubs	✗	✗	✓	✓
disassortativity	✗	✗	✗	✓
clustering	✗	✓	✗	✗
distances	✓	✗	✓	✓
all four at once	✗	✗	✗	✗

First Attempt

We built the obvious hybrid; it hit a wall

ddBA deletion (for disassortativity) + **Holme–Kim triangles** (for clustering)

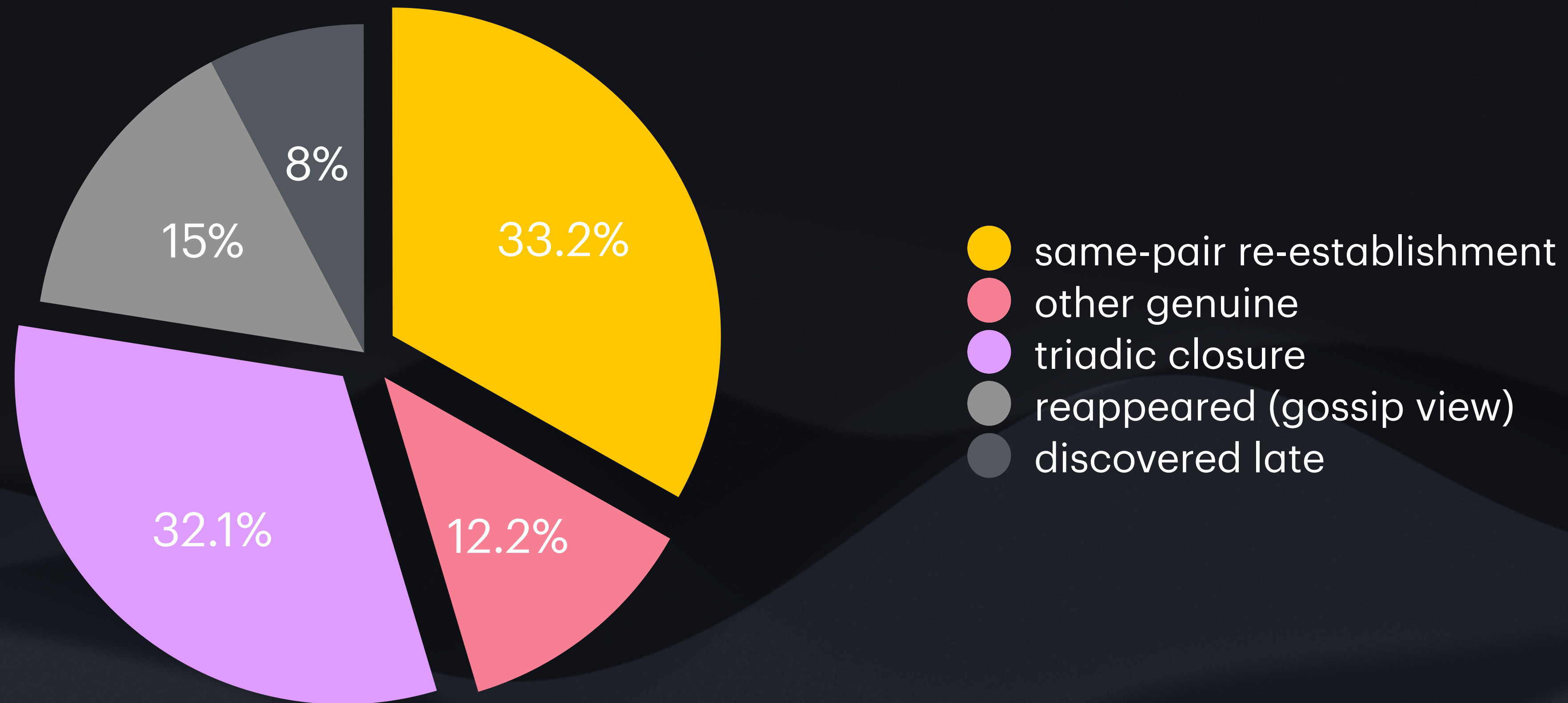
Three of four axes, then a wall: one triangle mechanism moves global and local clustering together. You match one, you lose the other.

Four knobs. Every knob fixed one number and ruined another.

Lesson: stop stacking mechanisms. Measure the network!

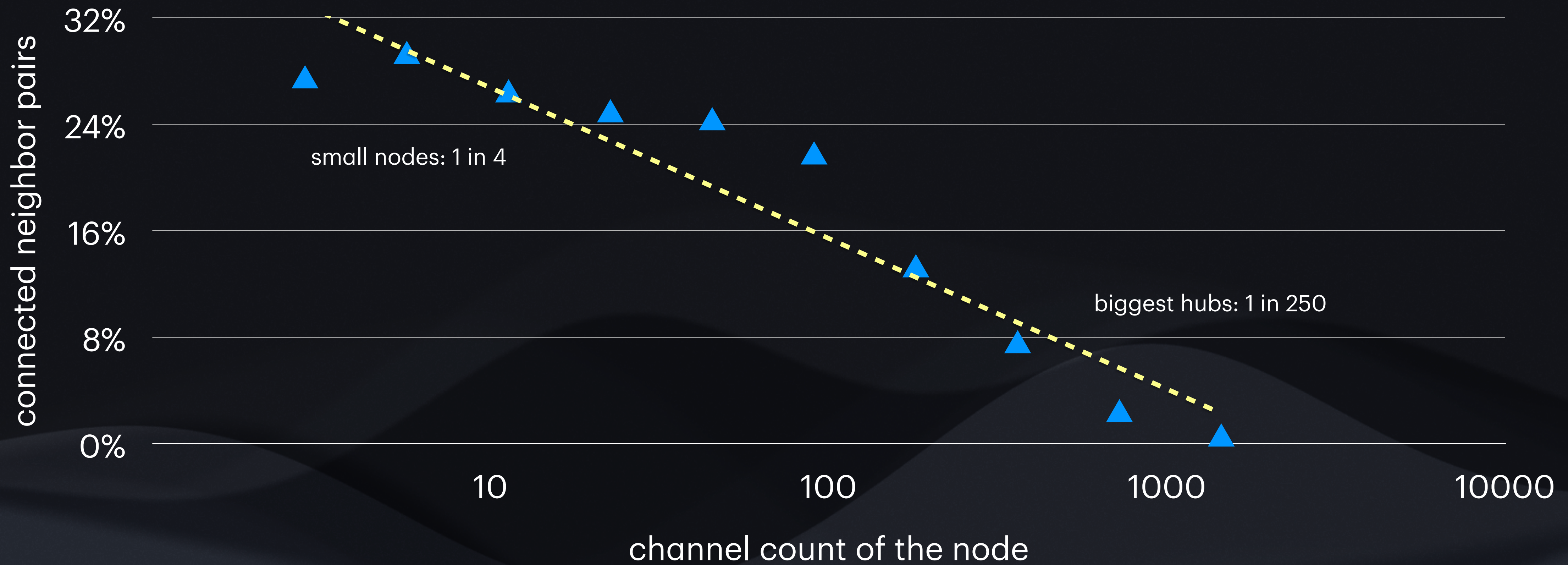
Measurement #1

What channel churn actually is



Measurement #2

Triangles form around small nodes, not hubs



Measurement #3

Hubs avoid hubs, leaves avoid leaves, but it's mostly arithmetic

$$r = -0.246$$

the real network: hubs connect to leaves, not to each other

$$r = -0.218$$

every partner reshuffled at random: only each node's channel count kept

$\approx 89\%$ of the effect is forced by the channel counts alone

A 1794-channel hub in a network that is 60% single-channel nodes has no choice: nearly every available partner is small. Preference explains only the last -0.028

The Model

Two changes to the classic Barabási-Albert recipe

Classic Barabási-Albert

$$d = m$$

every newcomer the same;
no leaves, hubs capped

$$\Pi(v) \propto k_v$$

strictly proportional to degree;
mixing fades to zero

Ours

$$P(d) \propto d^{-s}$$

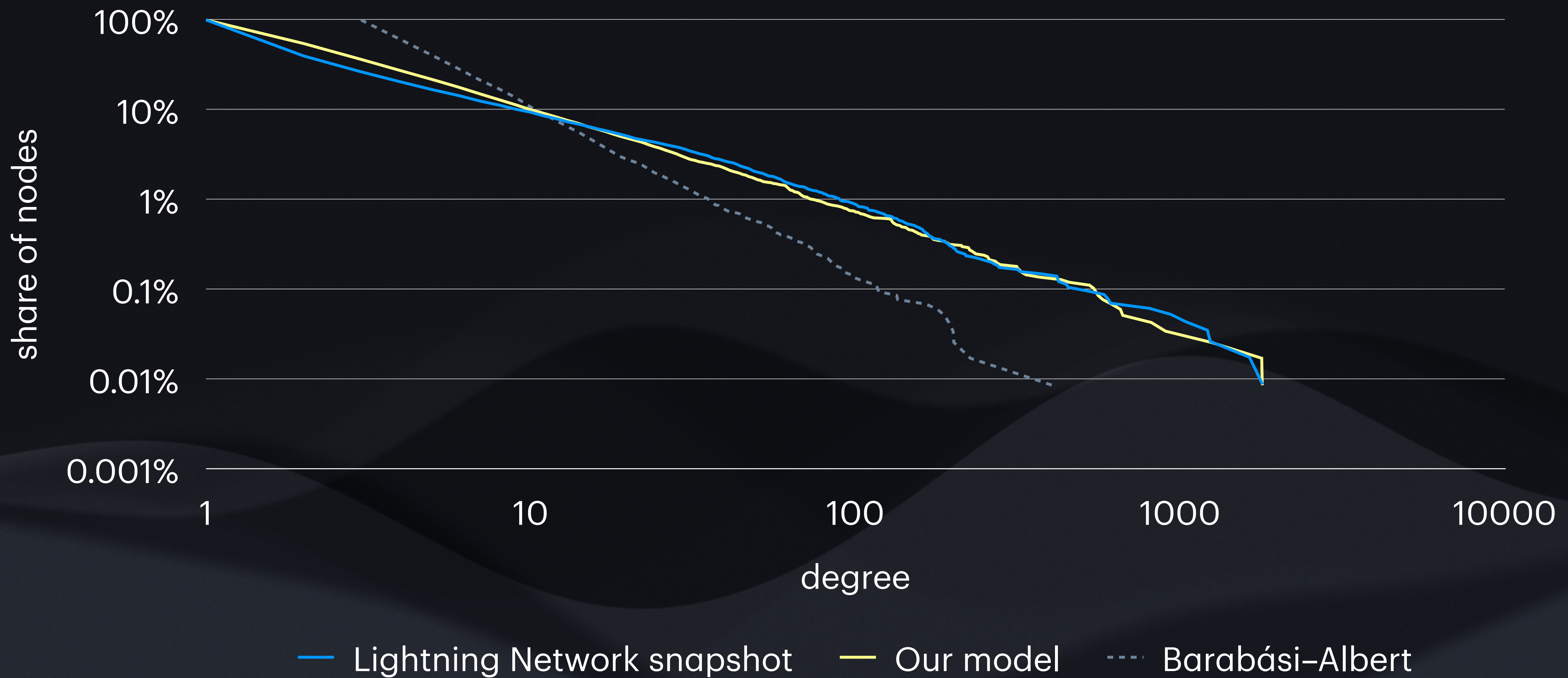
how many channels a newcomer opens;
cutoff follows from n

$$\Pi(v) \propto k_v^\alpha$$

whom a newcomer connects to;
bigger nodes slightly favored

Validation

Degree distribution, end to end



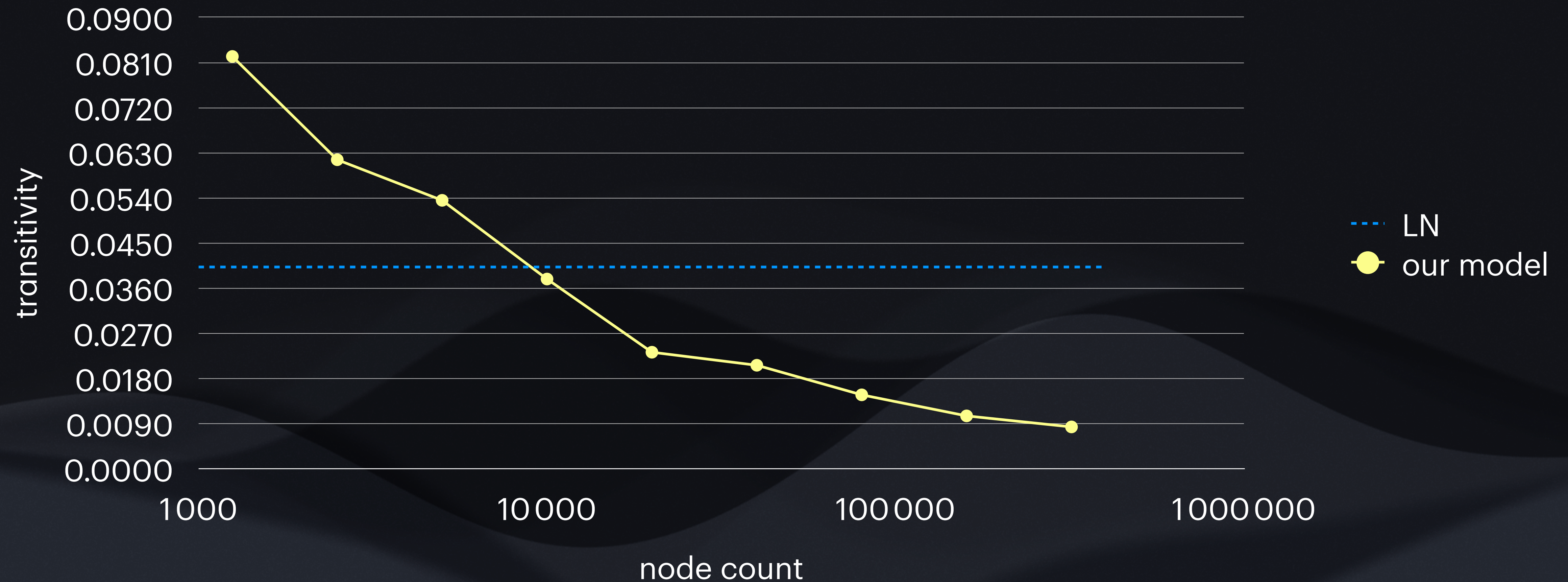
Validation

Closest on nearly every metric

metric	BA	ddBA	ours	LN
mean degree	6.00	4.99	6.00	6.33
share of degree-1 nodes	0%	17%	47%	60%
p99 degree	34	65	71	89
max degree	310	620	1896	1,794
transitivity	0.003	0.003	0.036	0.040
average local clustering	0.005	0.046	0.126	0.108
degree assortativity	-0.04	-0.24	-0.17	-0.25
average path length	4.33	3.75	3.40	3.59
bridges	0	1,865	5,724	6,893

The Fine Print

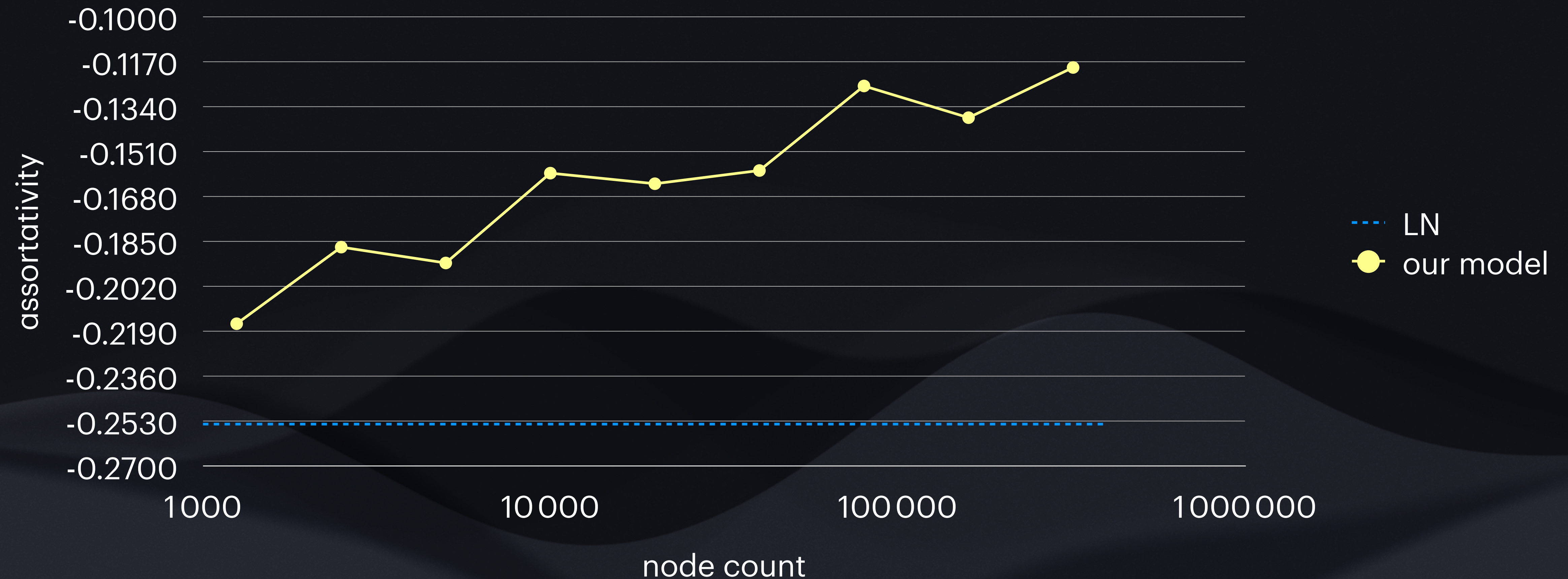
Gaps at today's size, forecasts at other sizes



transitivity falls as the network grows

The Fine Print

Gaps at today's size, forecasts at other sizes



assortativity weakens as the network grows

Maestro

<https://github.com/Bitstreams/Maestro>

The Laboratory

Maestro: run experiments on the graph you generated

```
schema_version: 1
name: learned_liquidity
...

topology:
  source: topology.graphml

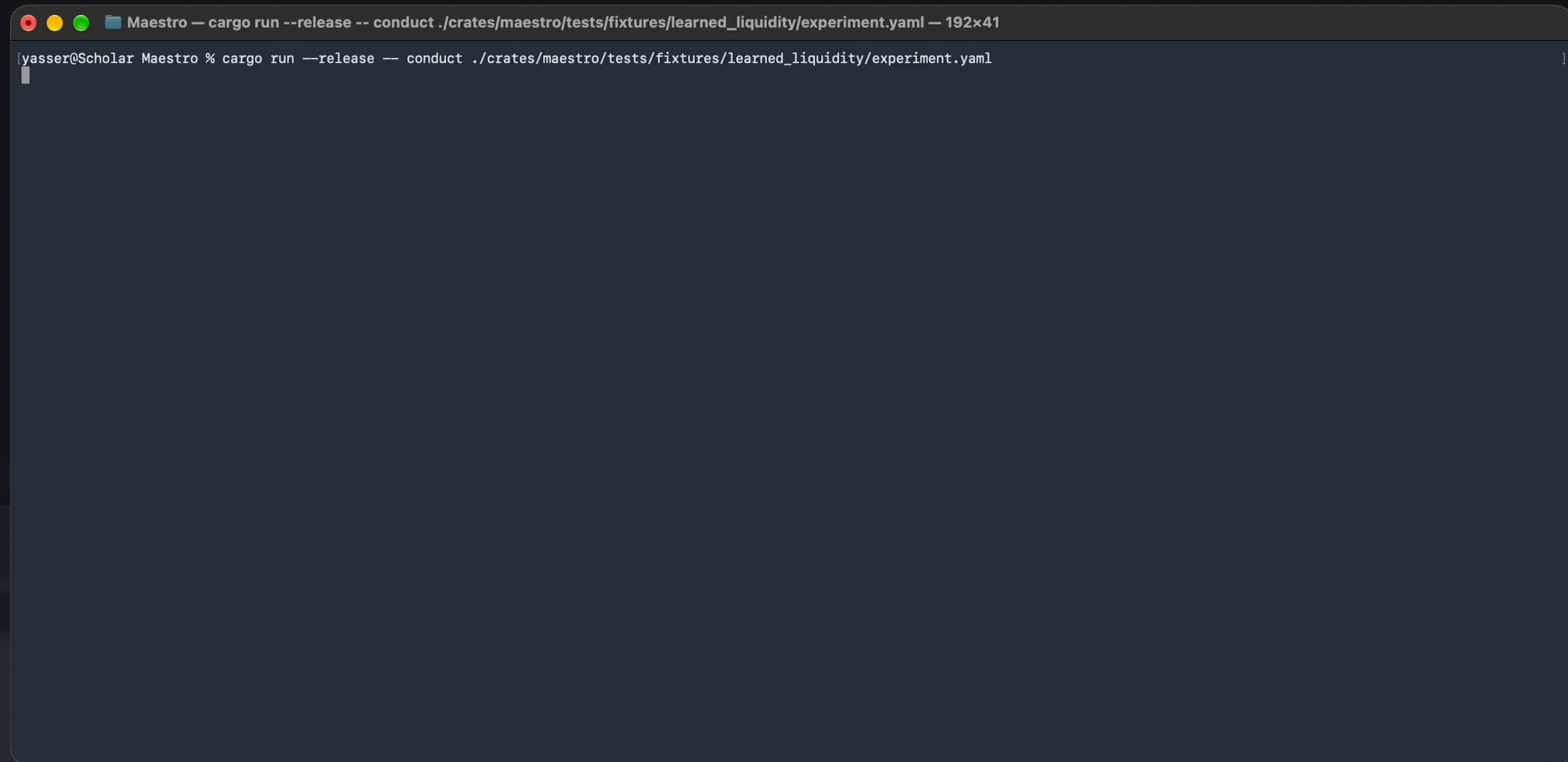
run:
  tasks:
    ...
    - name: lnd_n0_to_n4
      action:
        payment:
          source: n0
          destination: n4
          amount:
            constant: 800000000
            rate: 2 per 1s
            count: 5
```

```
<graph edgedefault="directed">
  <node id="n0"><data key="impl">lnd</data></node>
  <node id="n1"><data key="impl">cln</data></node>
  <node id="n2"><data key="impl">lnd</data></node>
  <node id="n3"><data key="impl">cln</data></node>
  <node id="n4"><data key="impl">lnd</data></node>

  <edge id="e0" source="n0" target="n1">
    <data key="cap">5000000000</data>
    <data key="bal">2500000000</data>
    <data key="fb">1000</data><data key="fppm">
      100
    </data>
    <data key="hmin">
      1000
    </data>
    <data key="hmax">4950000000</data>
    <data key="cltv">40</data>
    <data key="en">true</data>
  </edge>
```

The Demo

Maestro in action



```
Maestro — cargo run --release -- conduct ./crates/maestro/tests/fixtures/learned_liquidity/experiment.yaml — 192x41
[yasser@Scholar Maestro % cargo run --release -- conduct ./crates/maestro/tests/fixtures/learned_liquidity/experiment.yaml]
```

The Output

What Maestro captures

```
{ "timestamp": 182.992196917, "kind": "payment_attempted", "observer": "n0", "source": "n0", "destination": "n4",  
  "amount": 800000000, "attempt_identifier": "run.tasks.lnd_n0_to_n4.0" }  
{ "timestamp": 183.003228417, "kind": "invoice_created", "observer": "n4", "destination": "n4", "amount": 800000000,  
  "attempt_identifier": "run.tasks.lnd_n0_to_n4.0" }  
{ "timestamp": 183.07282125, "kind": "payment_attempted", "observer": "n0", "source": "n0", "destination": "n4",  
  "amount": 800000000, "attempt_identifier": "run.tasks.lnd_n0_to_n4.1" }  
{ "timestamp": 183.077387583, "kind": "invoice_created", "observer": "n4", "destination": "n4", "amount": 800000000,  
  "attempt_identifier": "run.tasks.lnd_n0_to_n4.1" }  
{ "timestamp": 183.103517125, "kind": "payment_attempted", "observer": "n0", "source": "n0", "destination": "n4",  
  "amount": 800000000, "attempt_identifier": "run.tasks.lnd_n0_to_n4.2" }  
{ "timestamp": 183.105576917, "kind": "invoice_created", "observer": "n4", "destination": "n4", "amount": 800000000,  
  "attempt_identifier": "run.tasks.lnd_n0_to_n4.2" }  
{ "timestamp": 183.489267333, "kind": "payment_attempted", "observer": "n0", "source": "n0", "destination": "n4",  
  "amount": 800000000, "attempt_identifier": "run.tasks.lnd_n0_to_n4.3" }  
{ "timestamp": 183.490466, "kind": "invoice_created", "observer": "n4", "destination": "n4", "amount": 800000000,  
  "attempt_identifier": "run.tasks.lnd_n0_to_n4.3" }  
{ "timestamp": 183.49606025, "kind": "payment_failed", "observer": "n0", "source": "n0", "destination": "n4", "amount": 800000000,  
  "reason": "remote error: payment status: FAILED; failure_reason:" }
```

Your results are only as
good as your model