# Learning Tumor Evolution With Diffusion Models

Siddharth Sabata, Russell Schwartz
Carnegie Mellon University

# Outline

Introduction

Methods

Results

Discussion

# Outline

**Introduction**

Methods

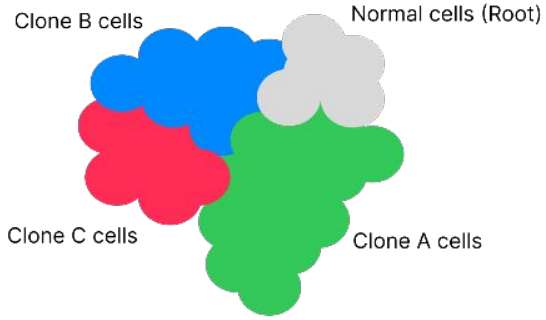
Results

Discussion

# Summary

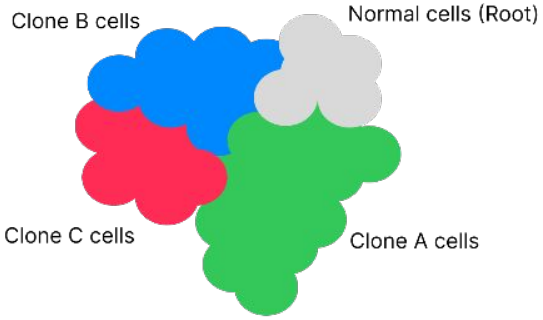
We successfully learned clonal phylogeny graph structures with a discrete diffusion model and simulated data

# Clonal Phylogenies

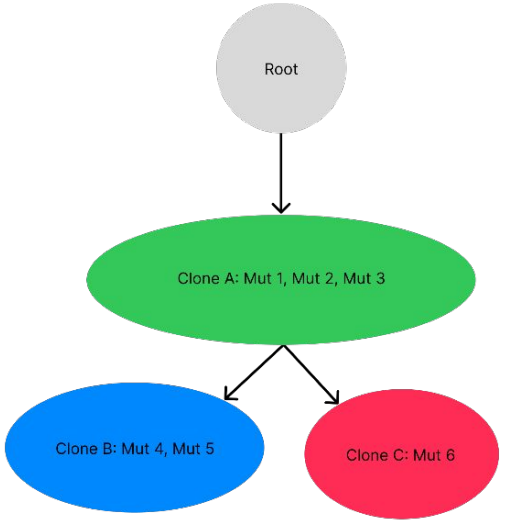



Clones: Tumor cell subpopulations

# Clonal Phylogenies




<u>Clones:</u> Tumor cell subpopulations




<u>Phylogenies:</u> Evolutionary trees

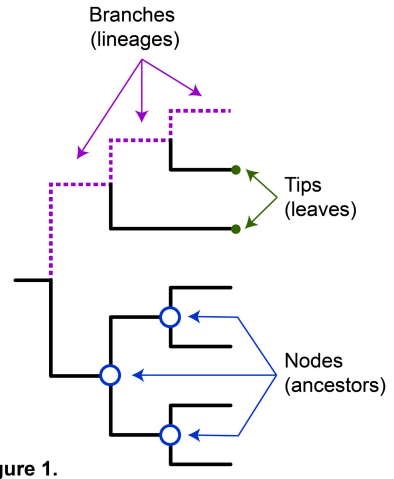
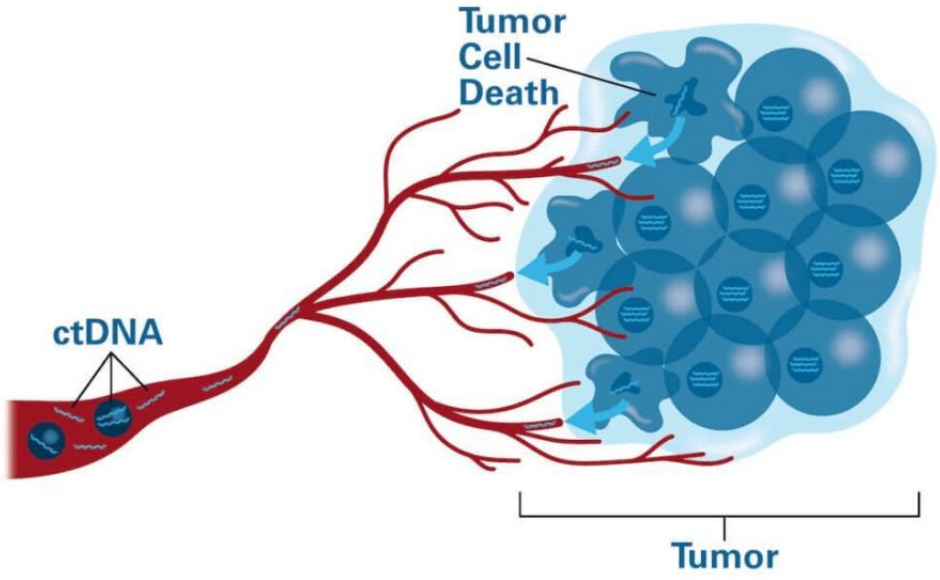



**Figure 1.**

https://amphibiaweb.org/taxonomy/

# Circulating Tumor DNA (ctDNA)




✅ Cheap and non-invasive

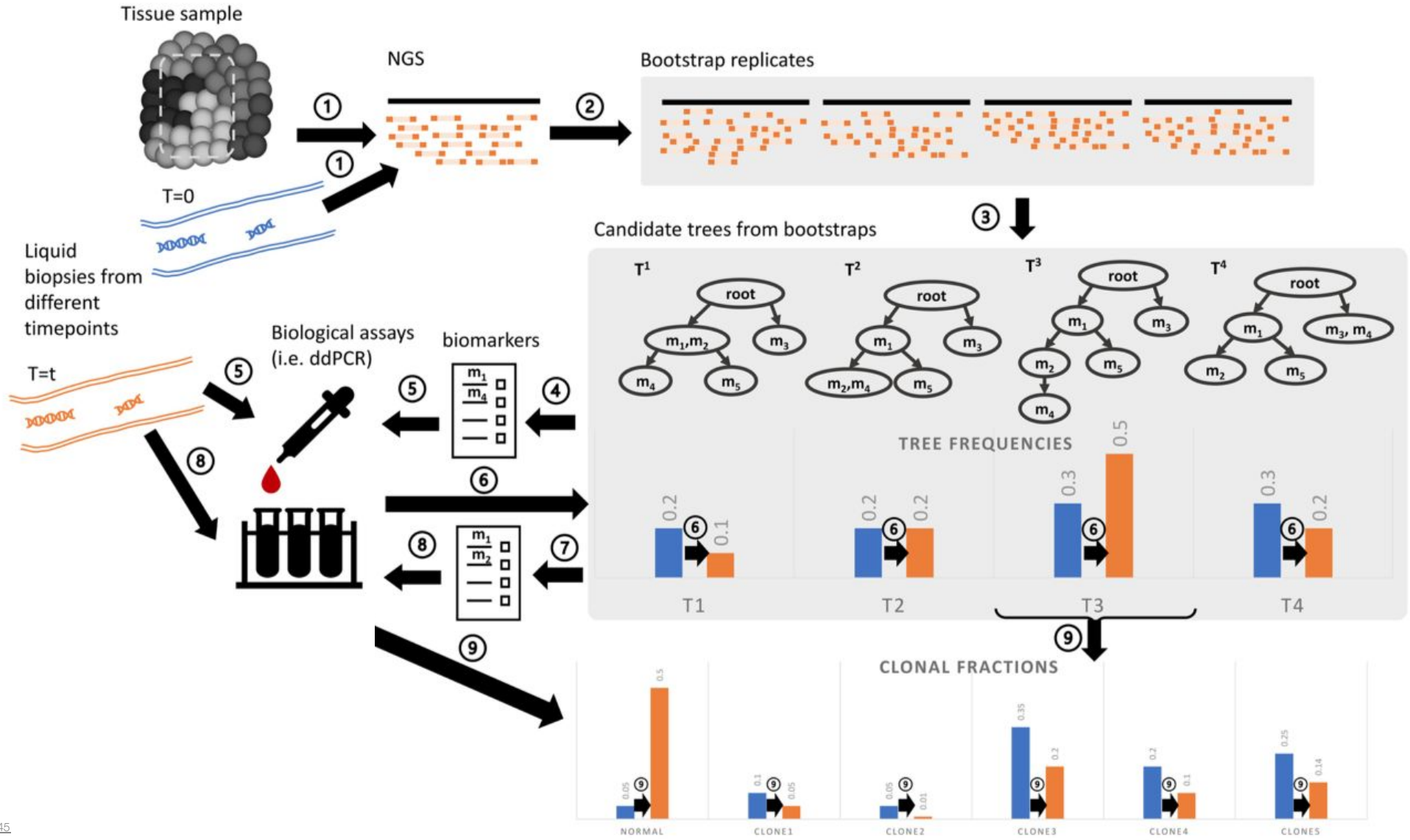
❓ Noisy and sparse

https://blog.dana-farber.org/insight/2023/06/what-is-ctdna-and-do-i-need-mine-tested/

# Clonal Phylogenies: Use Cases

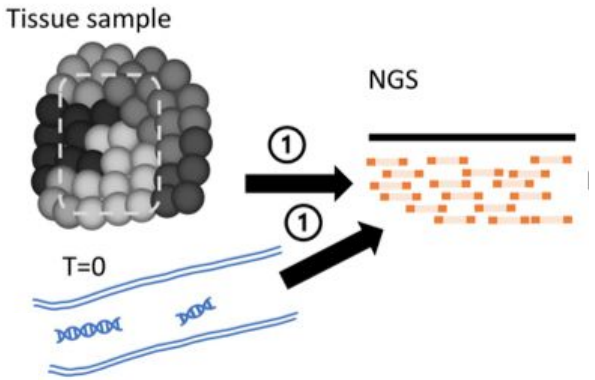
- Understanding tumor evolution → improved prediction
- Setting: Predicting patient relapse
- **Predict relapse with circulating tumor DNA (ctDNA)?**

# Use Case: Mase-phi



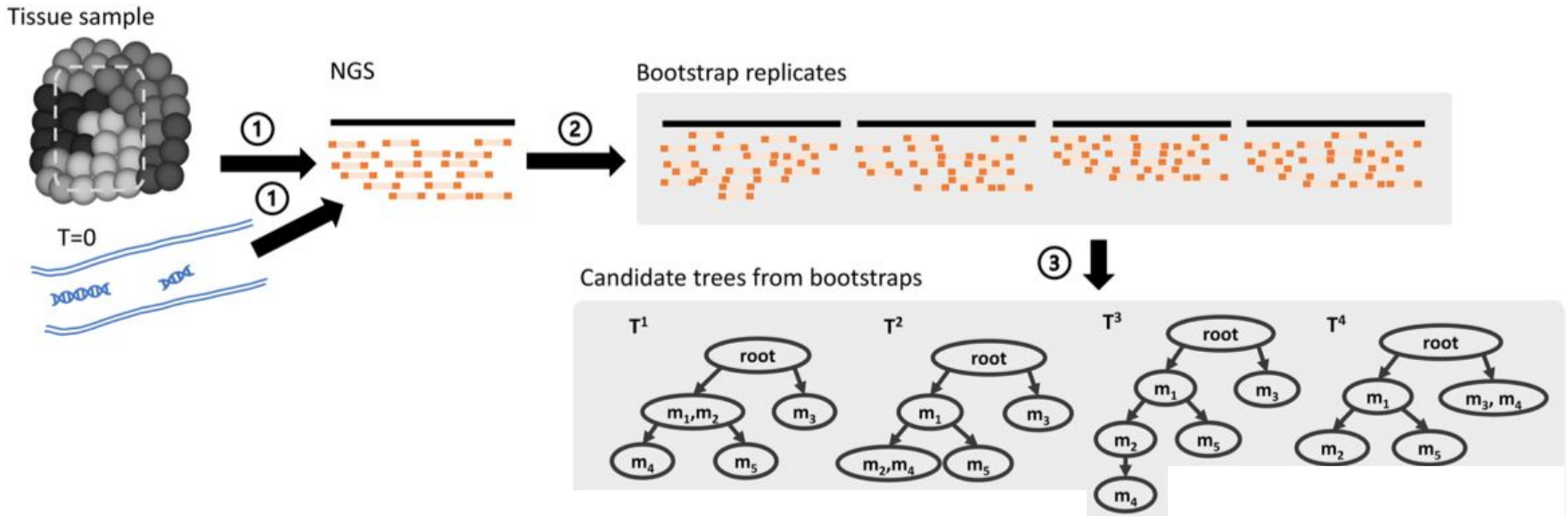

https://doi.org/10.1093/bioinformatics/btaf145

# Use Case: Mase-phi



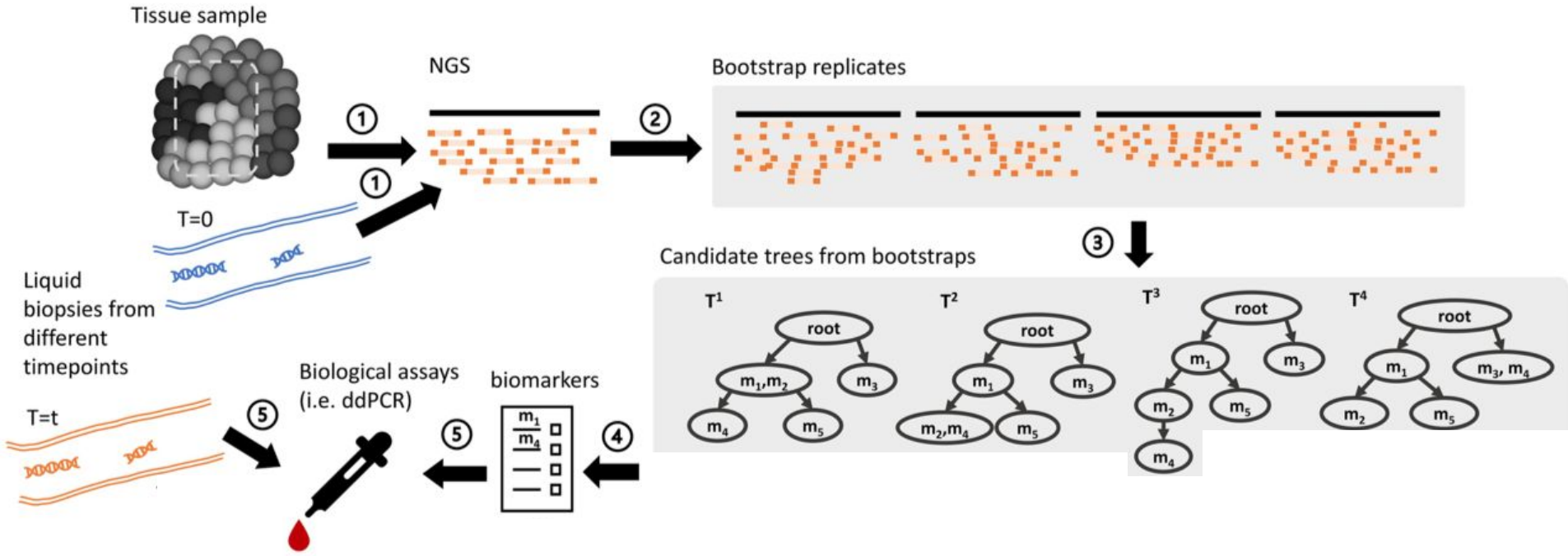

https://doi.org/10.1093/bioinformatics/btaf145

# Use Case: Mase-phi



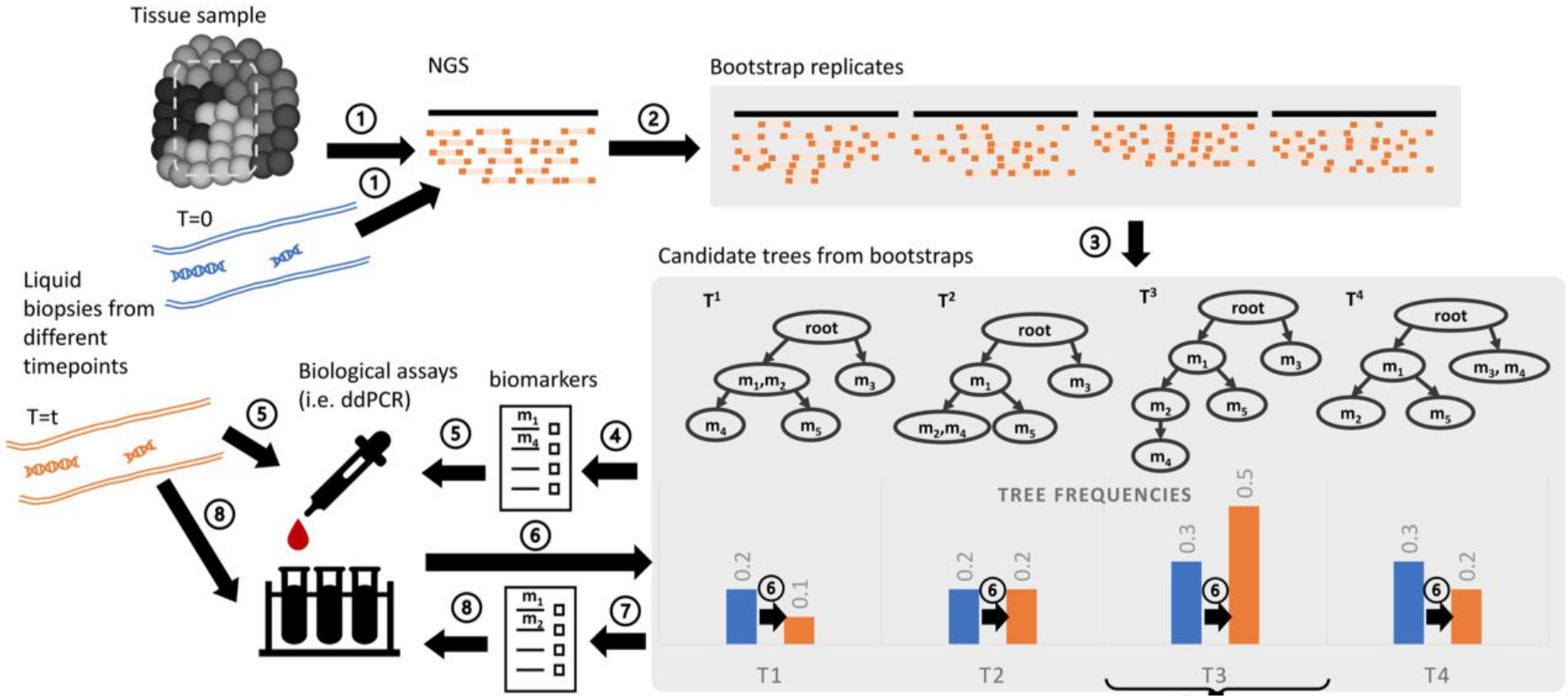

https://doi.org/10.1093/bioinformatics/btaf145

# Use Case: Mase-phi



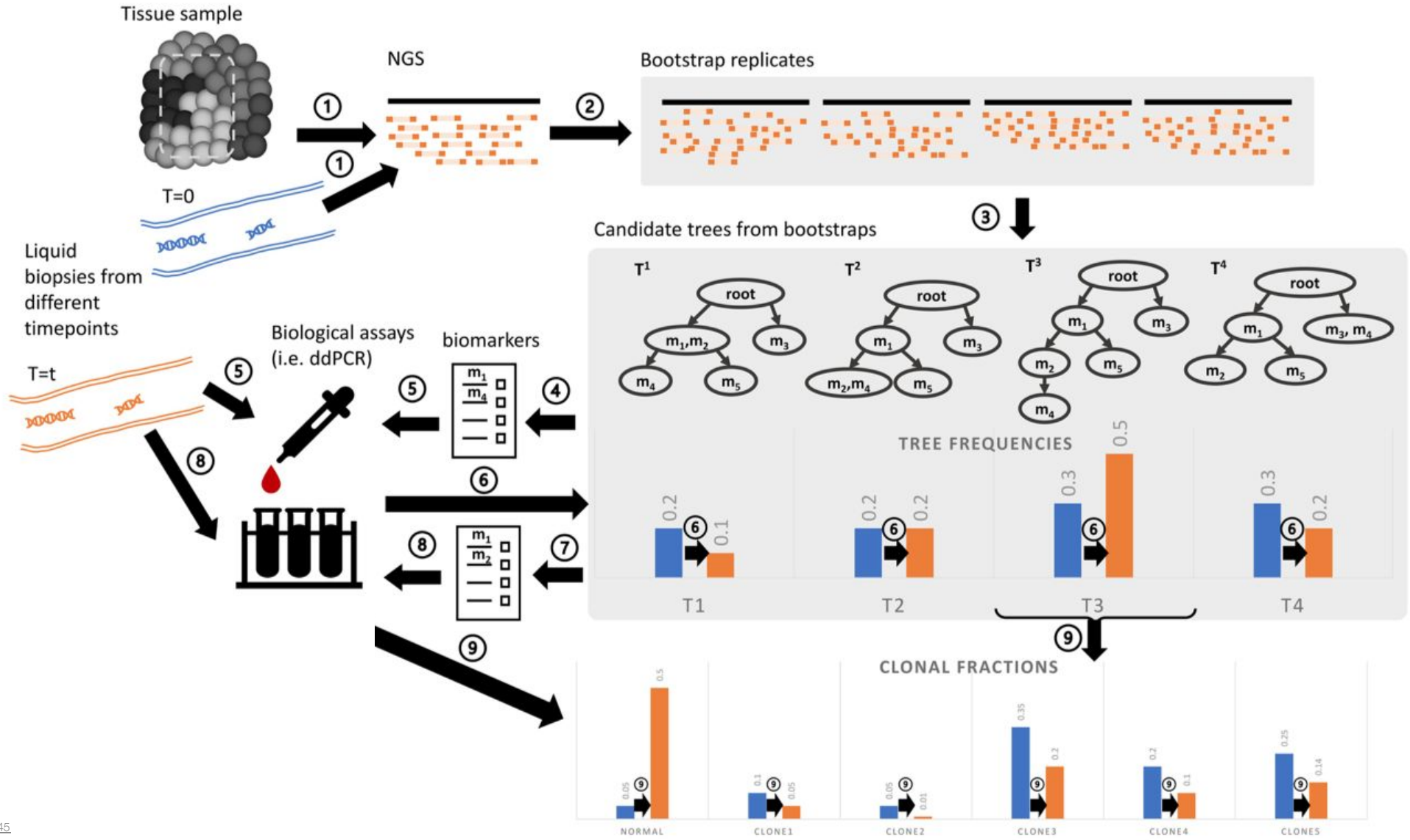

https://doi.org/10.1093/bioinformatics/btaf145

# Use Case: Mase-phi



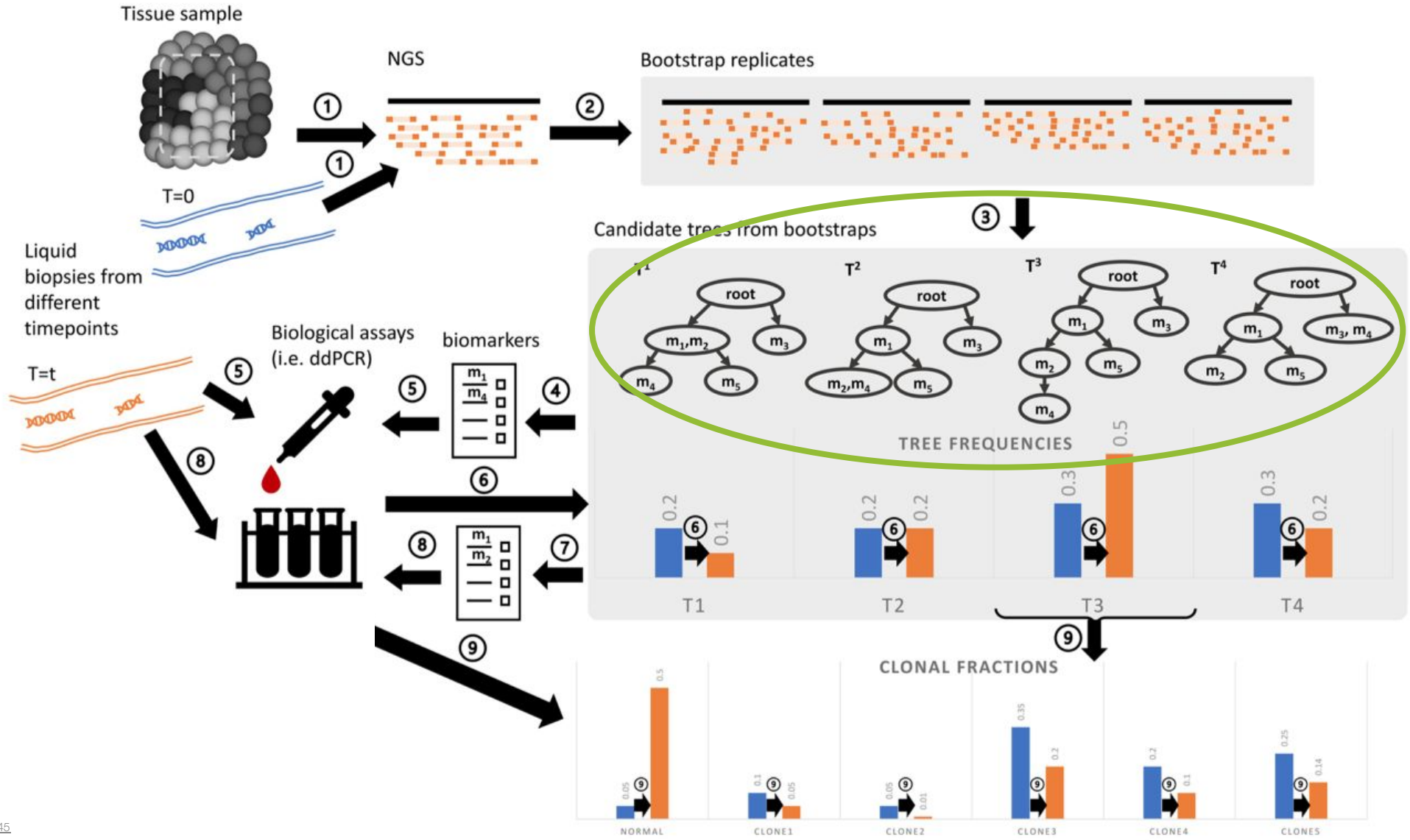

https://doi.org/10.1093/bioinformatics/btaf145

# Use Case: Mase-phi




https://doi.org/10.1093/bioinformatics/btaf145

# Phylogeny Distribution




https://doi.org/10.1093/bioinformatics/btaf145

# Phylogeny Reconstruction

f(read counts) → phylogeny tree

P(phylogeny | read counts)

- Current methods learn predefined parameters at **test-time** with Bayesian MCMC based approaches
- Instability → bootstrapping

# Related Prior Work

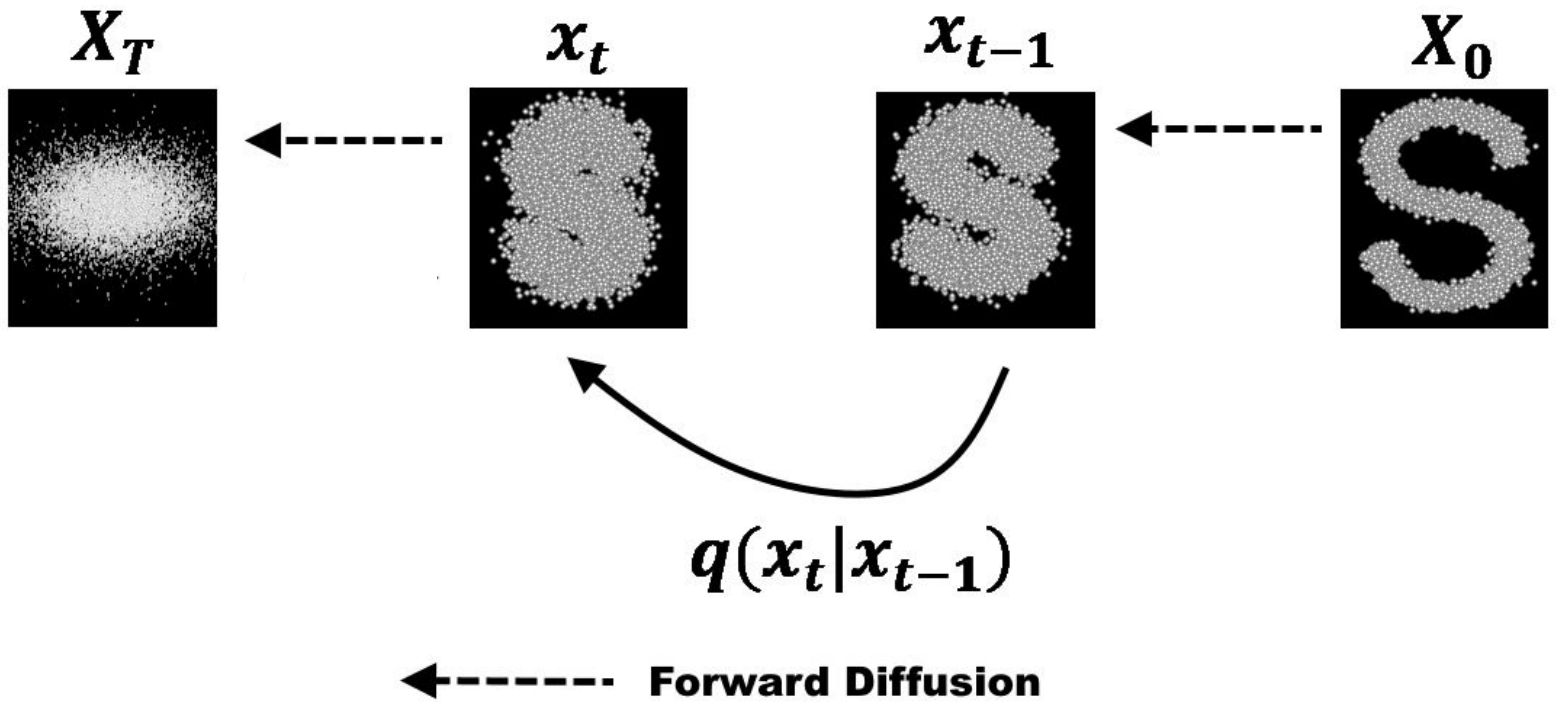
- Deep generative models use neural networks to learn and sample from **complex probability distributions**
- Some work has been done in applying generative modeling to learning phylogenies
  - bmVAE
  - PhyloVAE

https://doi.org/10.1093/bioinformatics/btac790
https://doi.org/10.48550/arXiv.2502.04730

# Diffusion Models

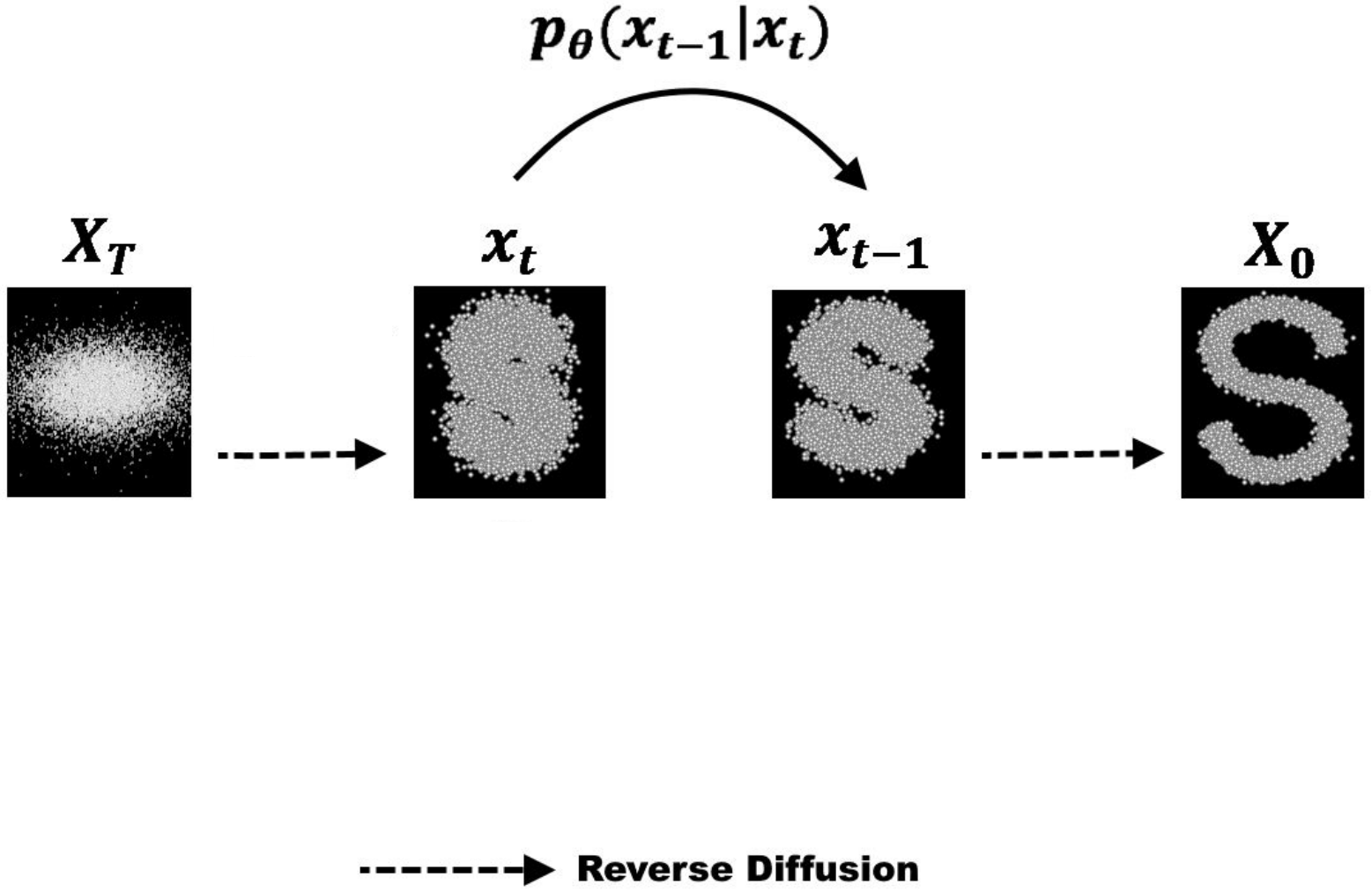
- Generative models great at **learning data distributions**
- Training-time optimization → faster inference
- Learn how to turn noise into a valid output

# Forward Diffusion



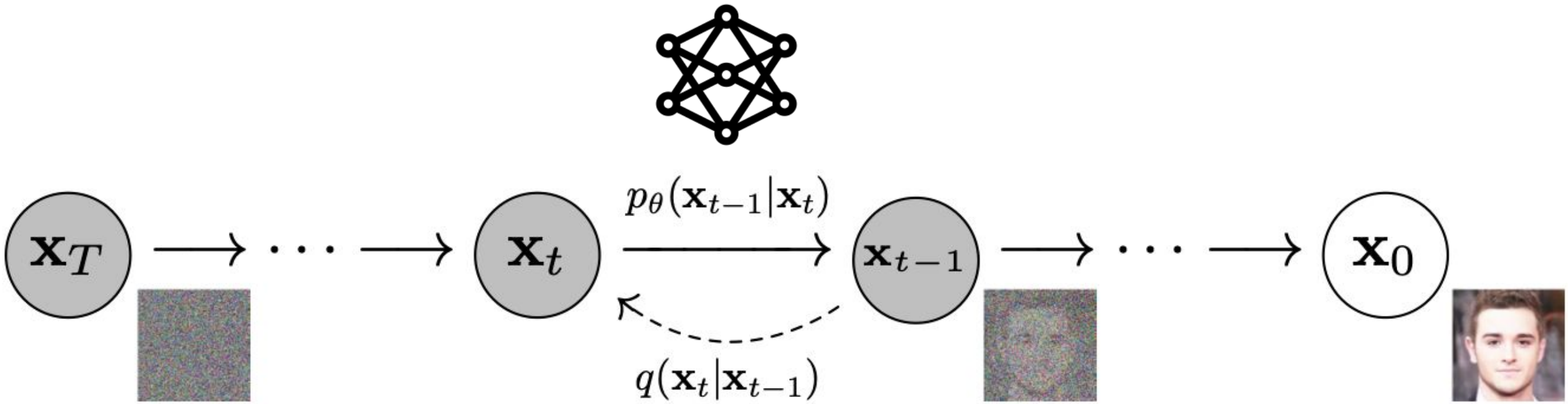

https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da/

# Reverse Diffusion




https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da/

# Training Diffusion Models




https://thenounproject.com/browse/icons/term/neural-network/
https://doi.org/10.48550/arXiv.2006.11239

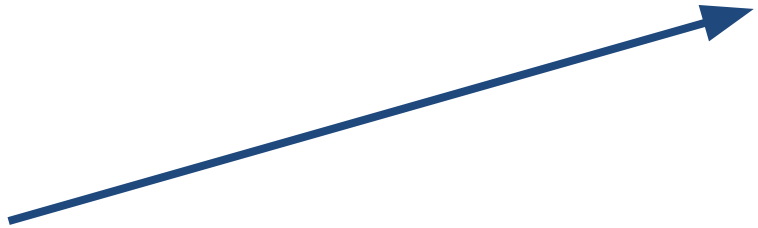
# Unconditional Diffusion

$X_T$

...

$p_\theta(X_{t-1} \mid X_t)$ $X_0$

# Unconditional Diffusion

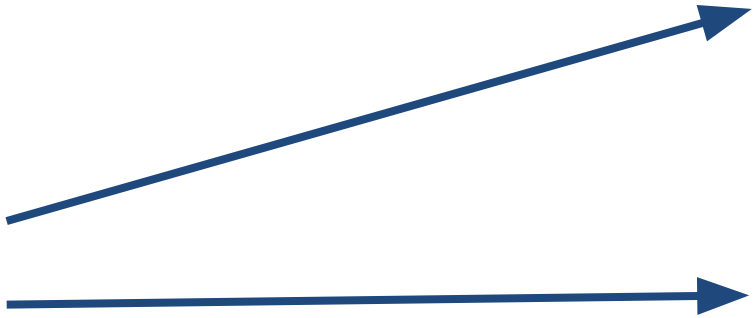


$X_T$

...

$p_\theta(X_{t-1} \mid X_t)$ $X_0$

# Unconditional Diffusion



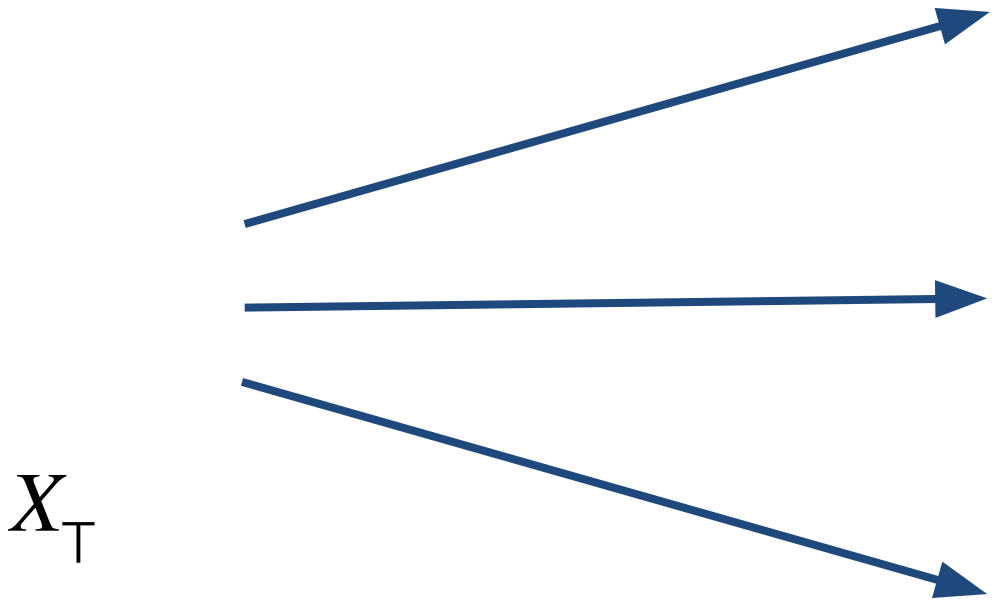
$X_T$

...

$p_\theta(X_{t-1} \mid X_t)$

$X_0$

# Unconditional Diffusion

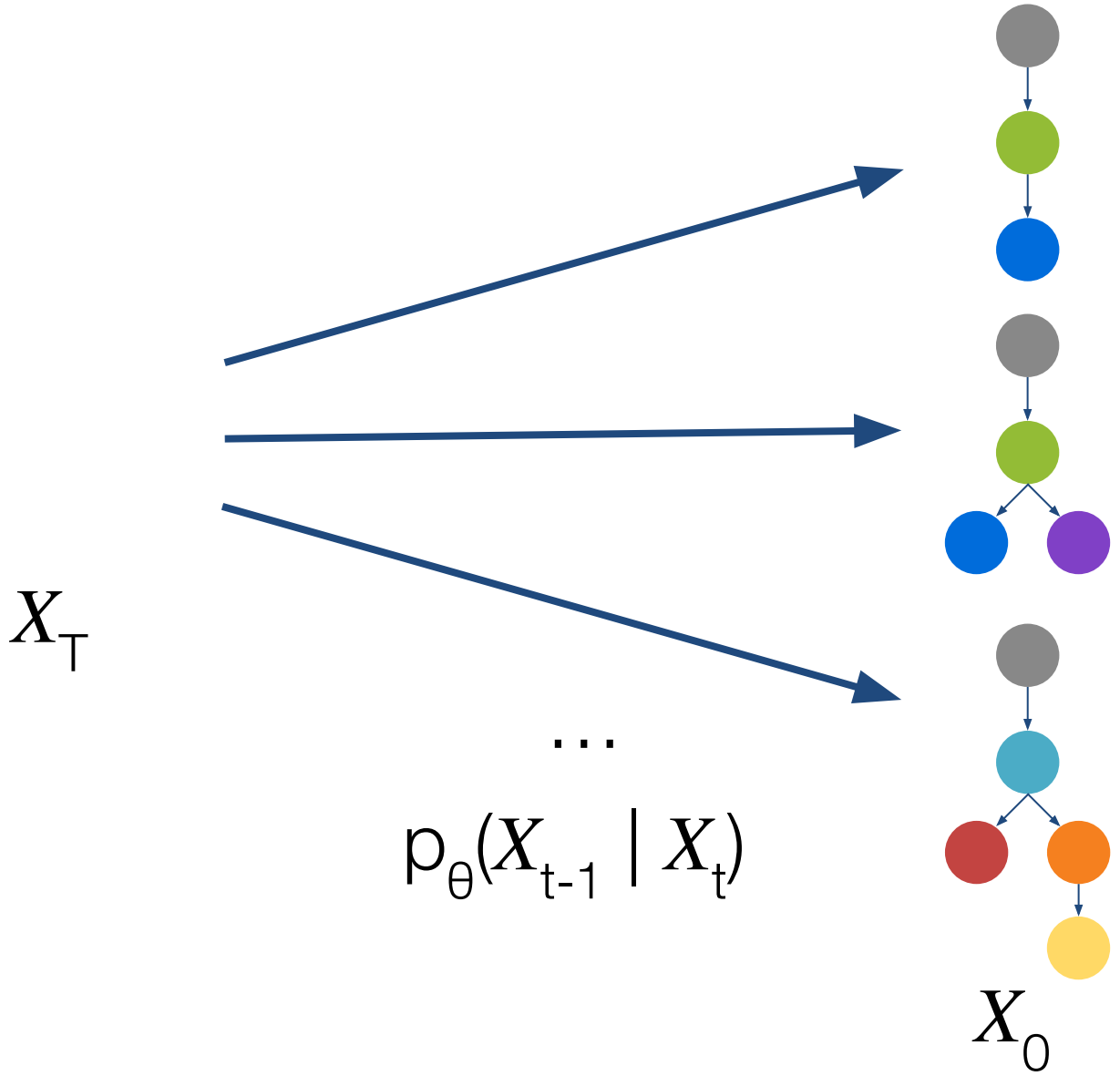


# Conditional Diffusion (Guidance)

y = “Santa”

$X_T$ $p_\theta(X_{t-1} \mid X_t, y)$ $X_0$

# Goal: Unconditional Phylogeny Generation



# Outline

Introduction

**Methods**

Results

Discussion

# Methods

Model selection

Training Data

Encoding

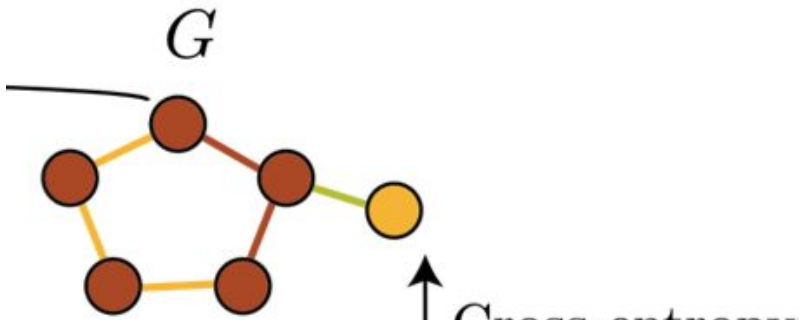
Training

# Model Selection: DiGress



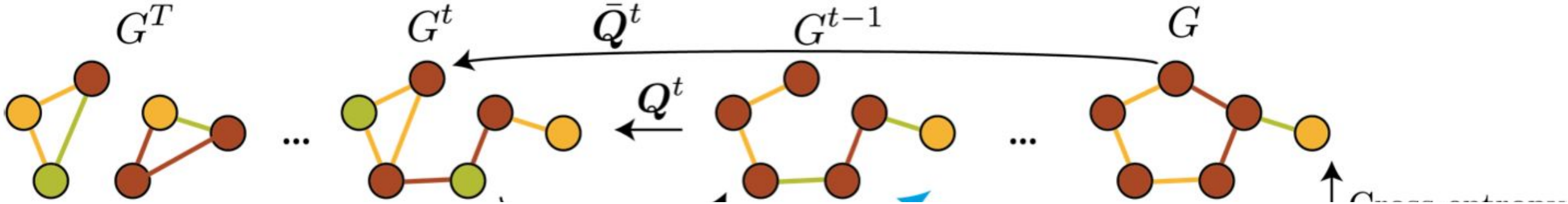
https://doi.org/10.48550/arXiv.2209.14734

# Model Selection: DiGress



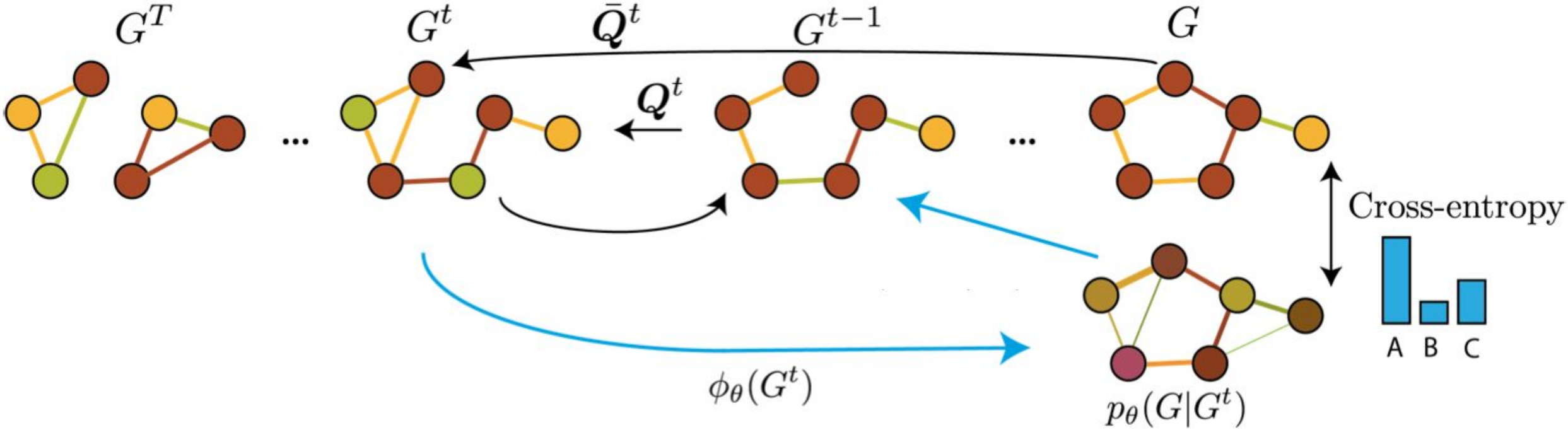

https://doi.org/10.48550/arXiv.2209.14734

# Model Selection: DiGress




https://doi.org/10.48550/arXiv.2209.14734

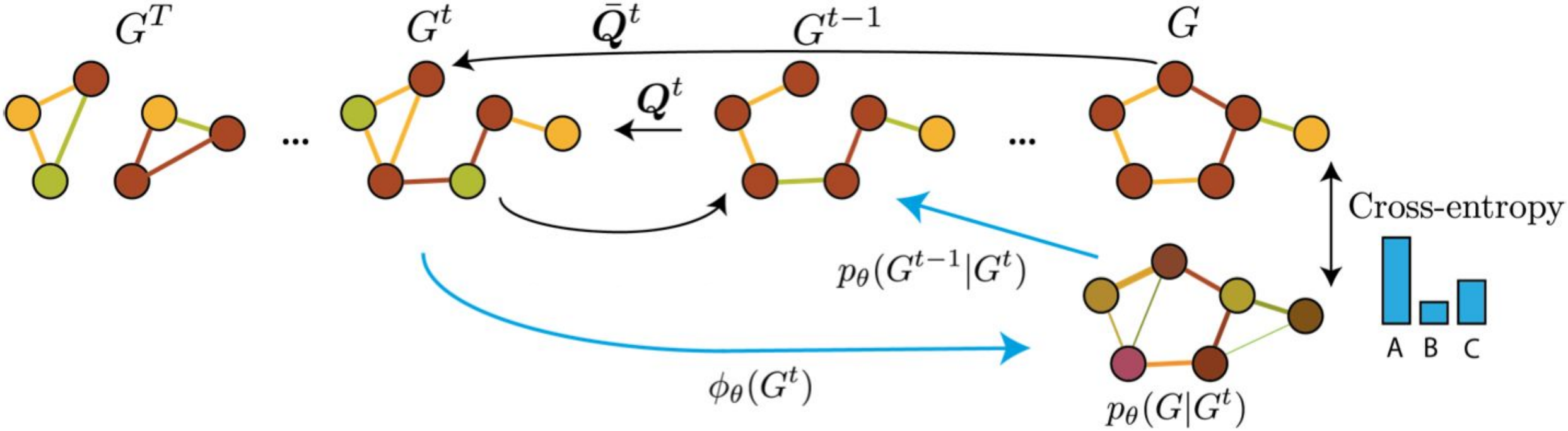
# Model Selection: DiGress





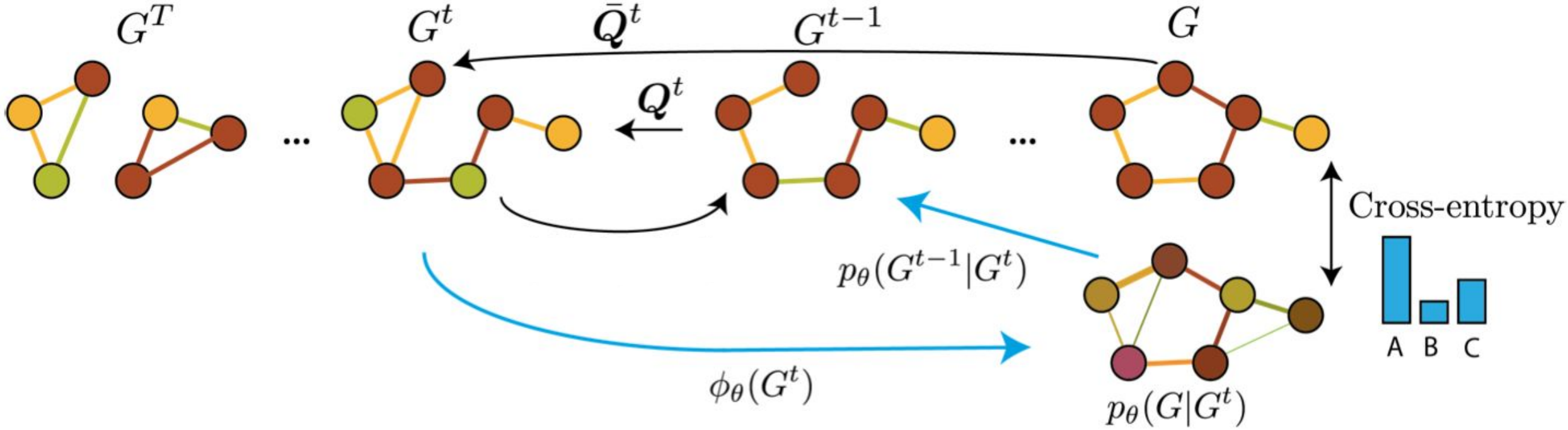

https://doi.org/10.48550/arXiv.2209.14734

# Model Selection: DiGress




https://doi.org/10.48550/arXiv.2209.14734

# Model Selection: DiGress



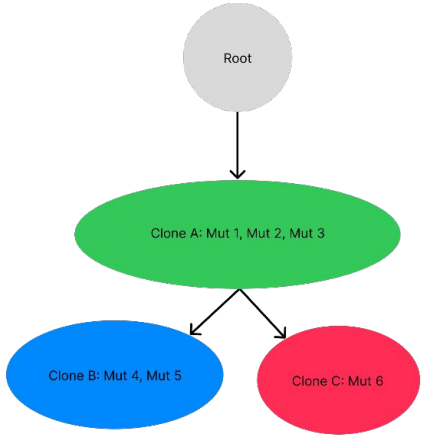

- Graph transformer based implementation
- Provides generalized framework we can adapt

https://doi.org/10.48550/arXiv.2209.14734

# Training Data

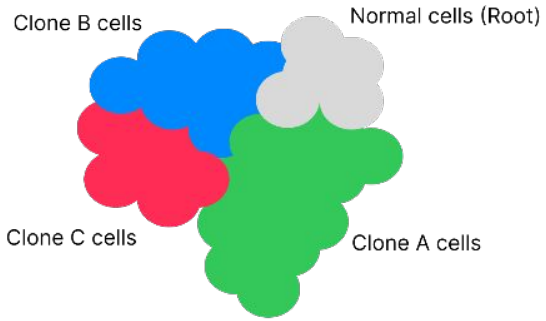
- Ground truth tumor phylogenies from nature are impossible → **simulated data**
- SISTEM simulates tumor growth, metastasis, and DNA sequencing
  - 1: **Growth** until minimum detectable population size
  - 2: **Sample** cells and sequence

# Training Data




- Provides ground truth trees paired with bulk sequencing read counts
  - Only ground truth trees used in this work
  - Paired read counts required for conditional generation

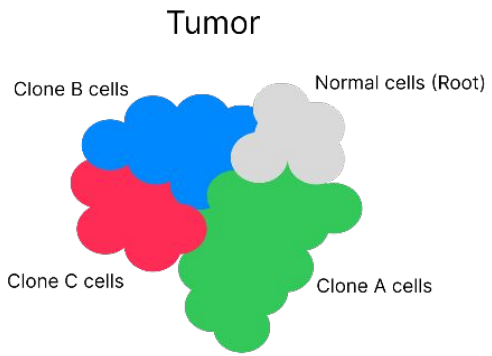
# Training Data



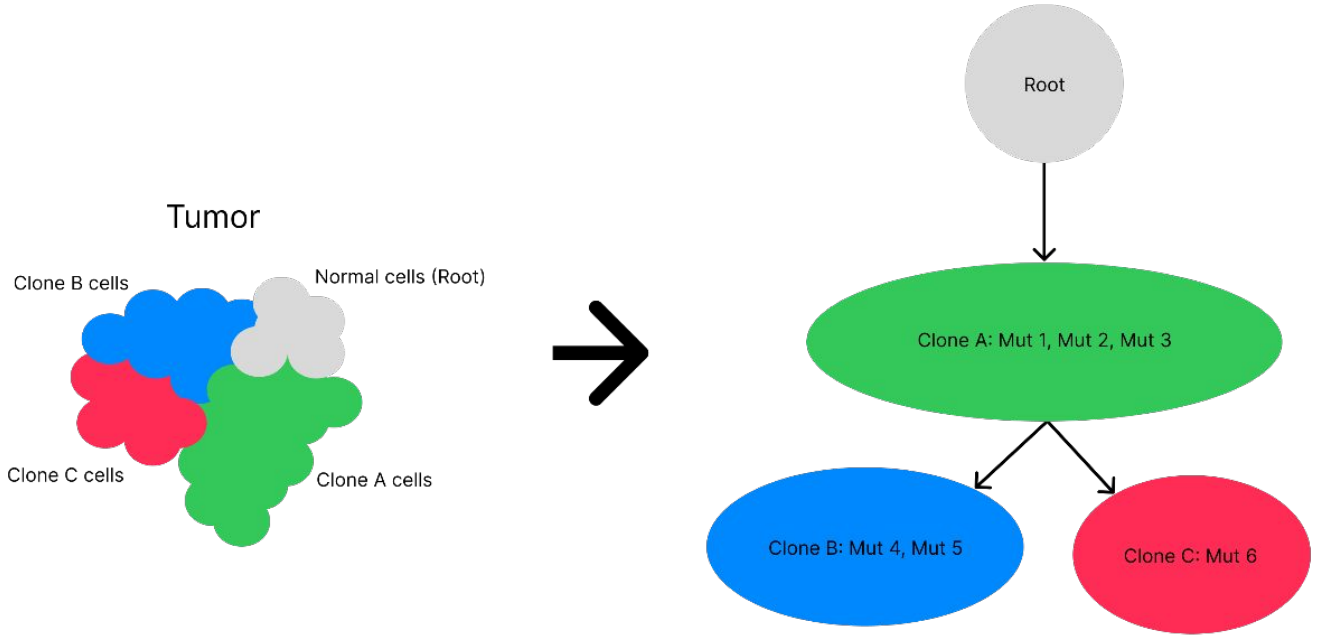

1,967 tumor phylogenies used for training

- One single parameter set
- 800 unique tumors
- 3 resamples for each

# Encoding

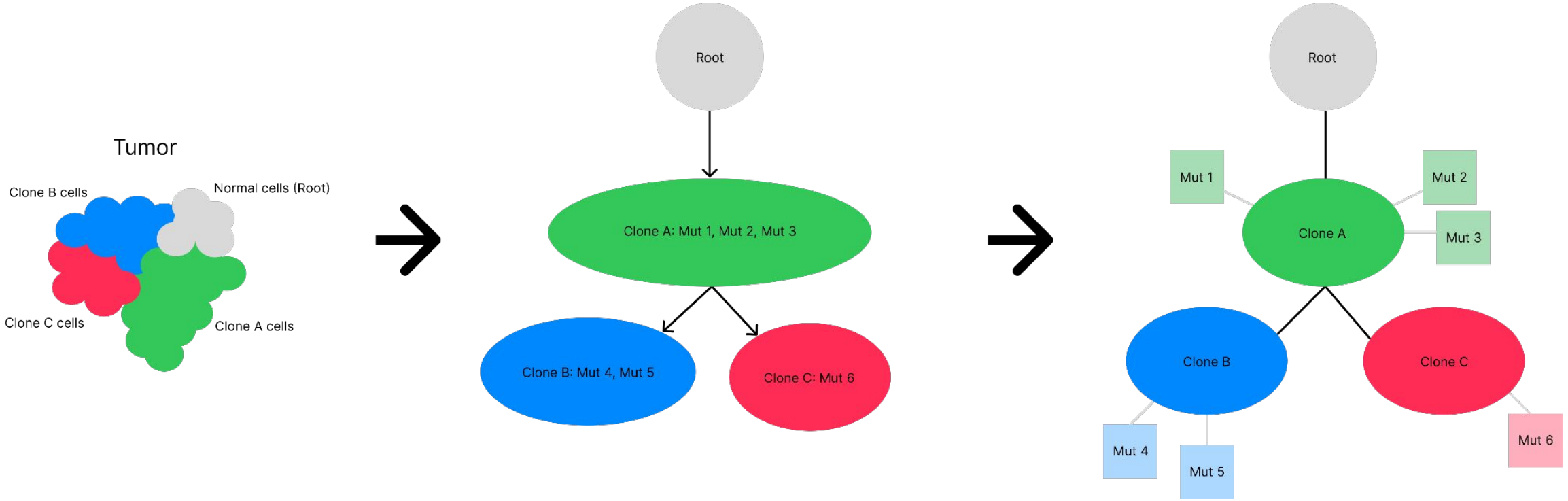

Tumor
Clone B cells
Normal cells (Root)
Clone C cells
Clone A cells

# Encoding



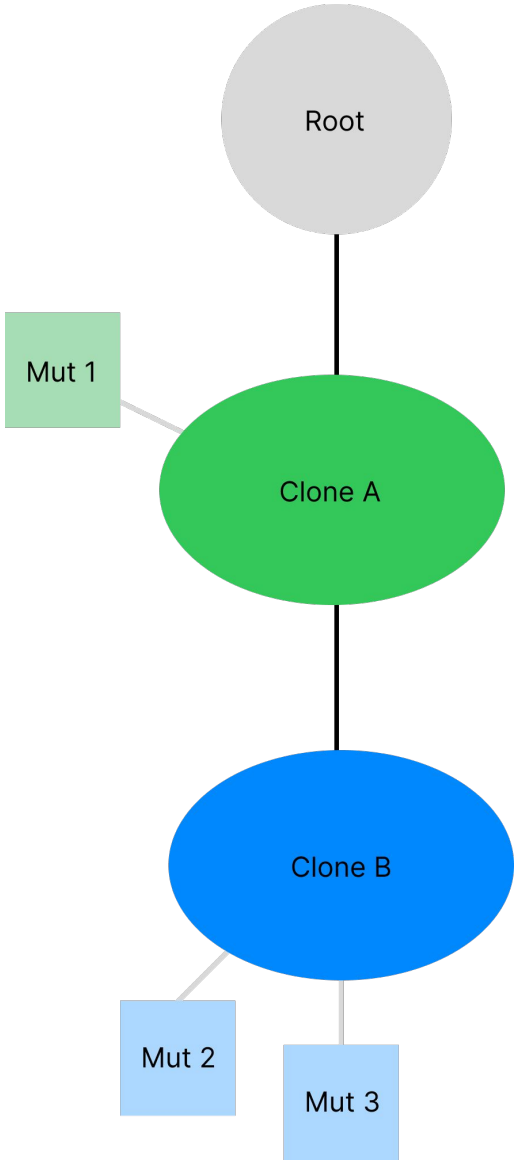

- DiGress supports graphs with **undirected edges** and **single class nodes**

# Encoding




- DiGress supports graphs with **undirected edges** and **single class nodes**

# Node Encoding




$$X = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

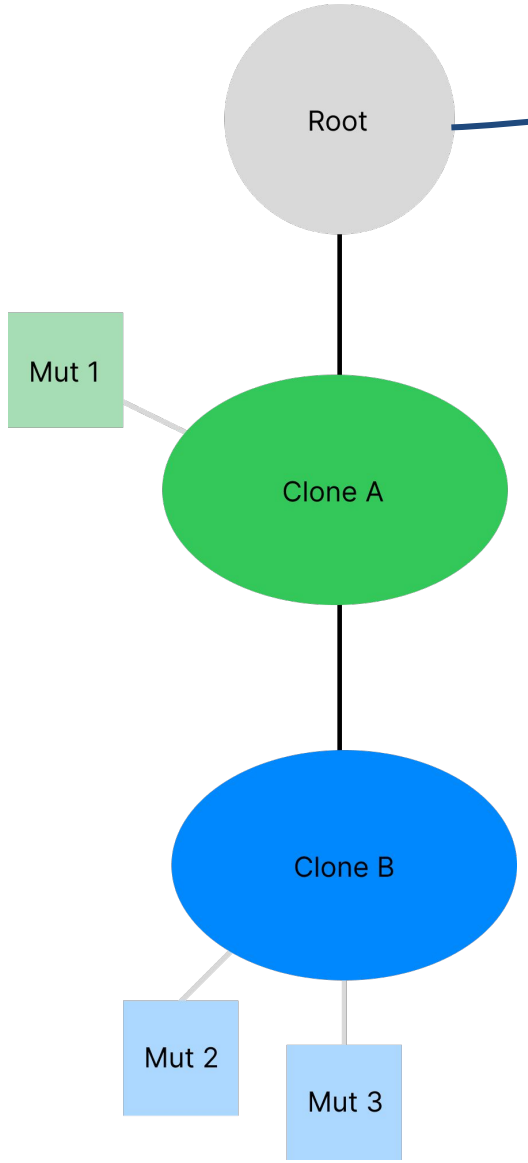
Node Types

0: Root

1: Clone

2: Mutation

# Node Encoding




$$X = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

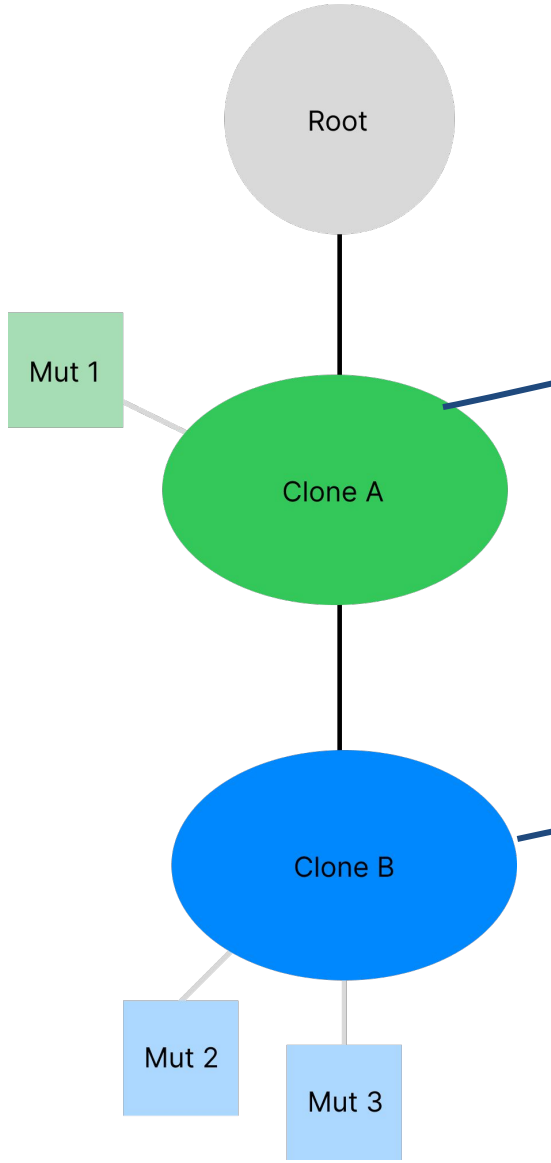
Node Types

**0: Root**

1: Clone

2: Mutation

# Node Encoding




$$X = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

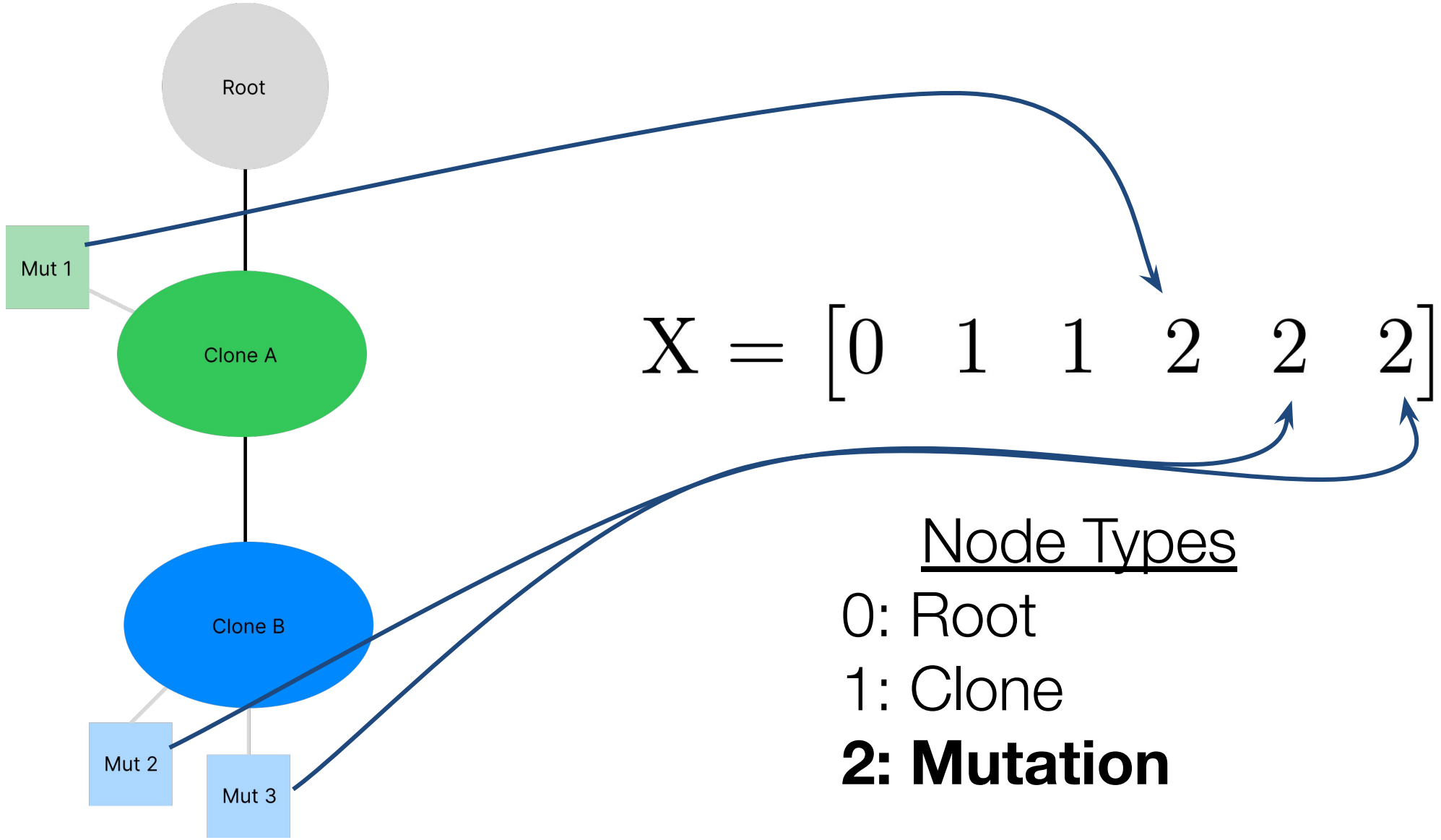
Node Types

0: Root

**1: Clone**

2: Mutation

# Node Encoding




$$X = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

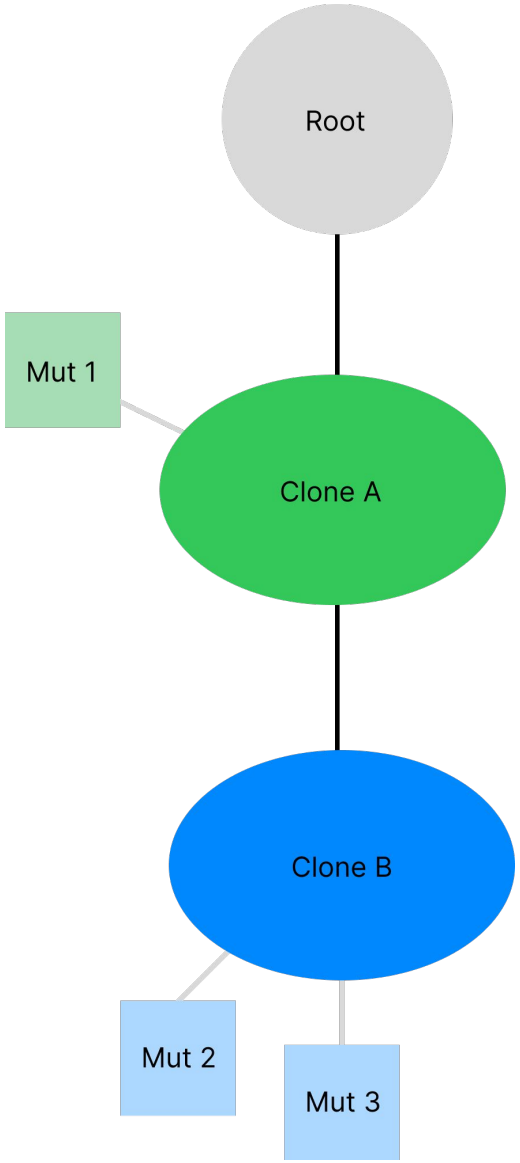
Node Types

0: Root

1: Clone

**2: Mutation**

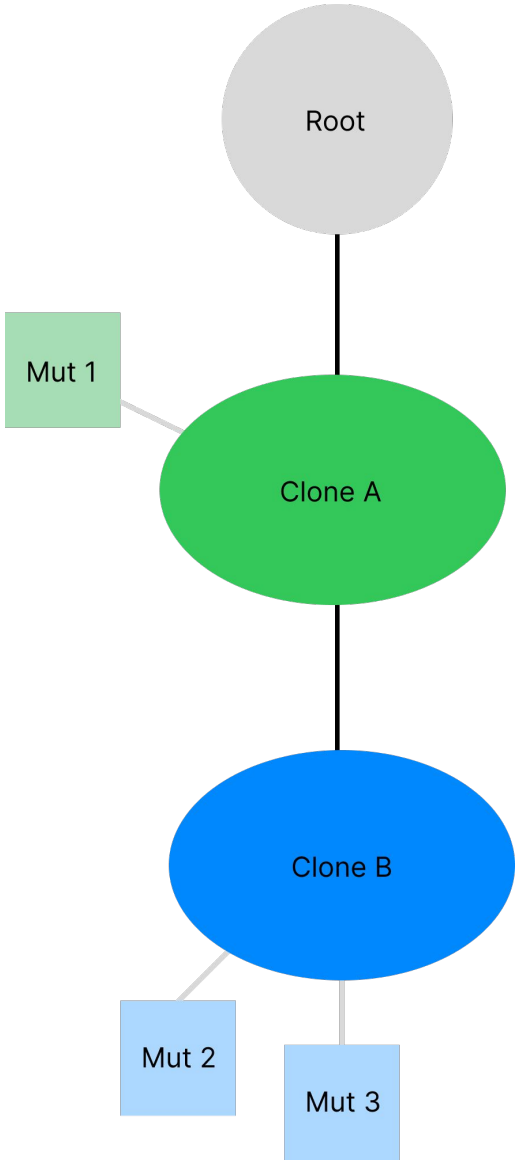
# Edge Encoding




$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

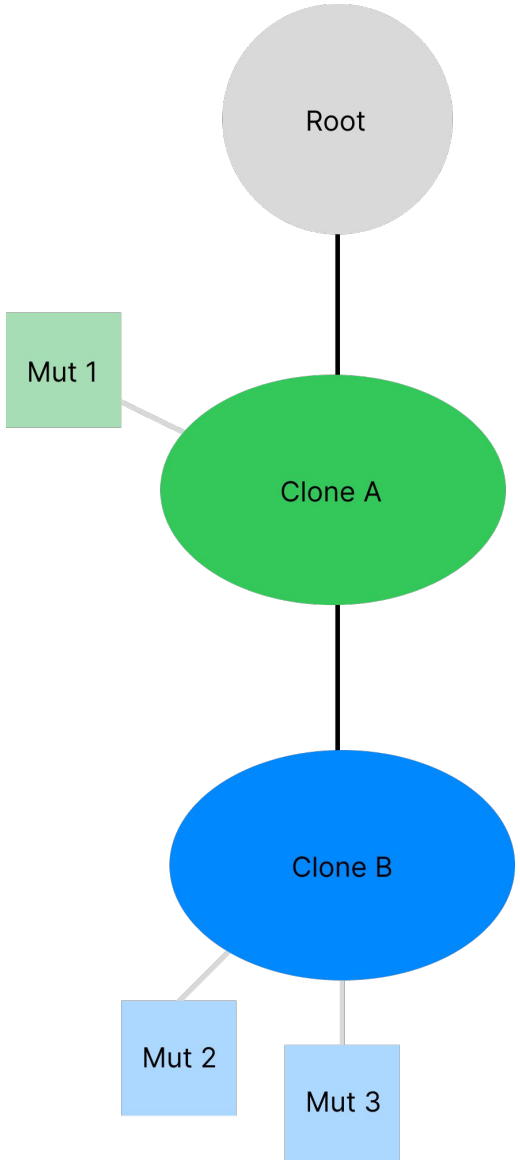
# Edge Encoding




$$E = \begin{matrix} \text{Root} \\ \text{Clone A} \\ \text{Clone B} \\ \text{Mut 1} \\ \text{Mut 2} \\ \text{Mut 3} \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$X = \begin{matrix} \text{Root} & \text{Clone A} & \text{Clone B} & \text{Mut 1} & \text{Mut 2} & \text{Mut 3} \end{matrix} \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

# Edge Encoding




Edge Types

0: None

1: Clone

2: Mutation

$$E = \begin{array}{c} \text{Root} \\ \text{Clone A} \\ \text{Clone B} \\ \text{Mut 1} \\ \text{Mut 2} \\ \text{Mut 3} \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$X = \begin{array}{cccccc} \text{Root} & \text{Clone A} & \text{Clone B} & \text{Mut 1} & \text{Mut 2} & \text{Mut 3} \end{array} \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

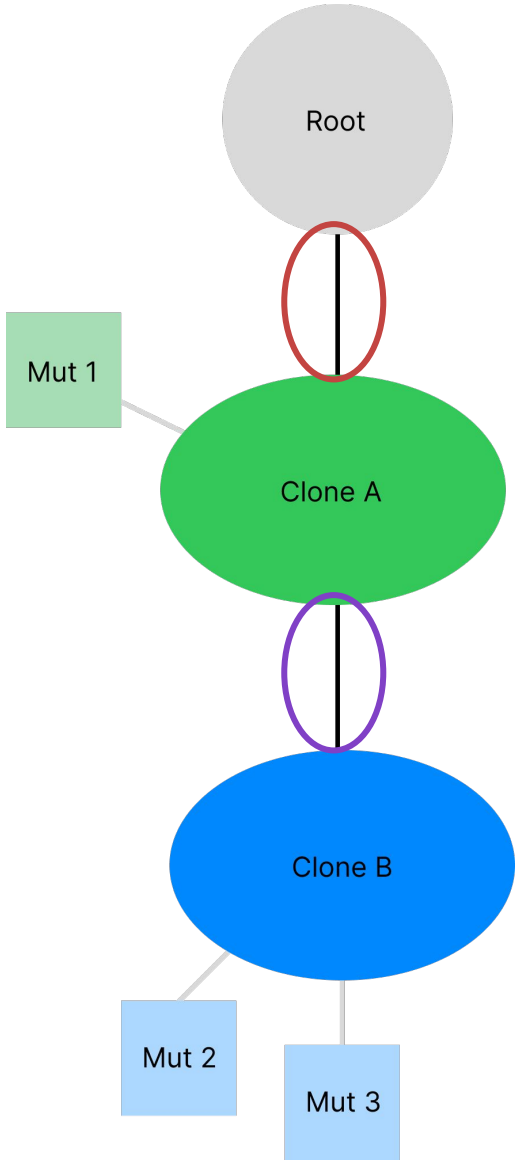
# Edge Encoding

Edge Types

0: None

**1: Clone**

2: Mutation




$$E = \begin{array}{c} \\ \text{Root} \\ \text{Clone A} \\ \text{Clone B} \\ \text{Mut 1} \\ \text{Mut 2} \\ \text{Mut 3} \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$X = \begin{array}{c} \text{Root} & \text{Clone A} & \text{Clone B} & \text{Mut 1} & \text{Mut 2} & \text{Mut 3} \\ \end{array} \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

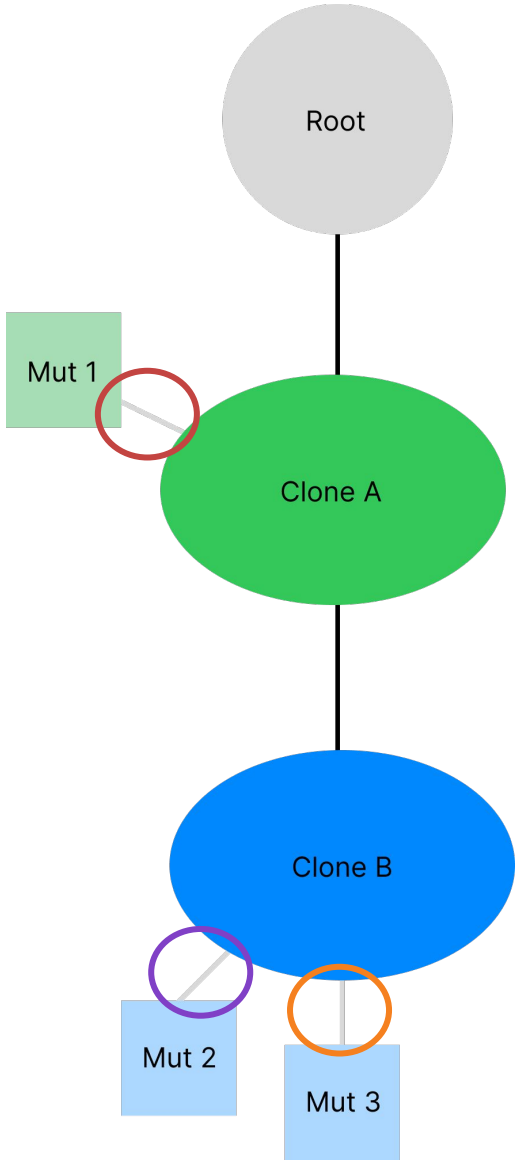
# Encoding Tensor

Edge Types

0: None

1: Clone

**2: Mutation**



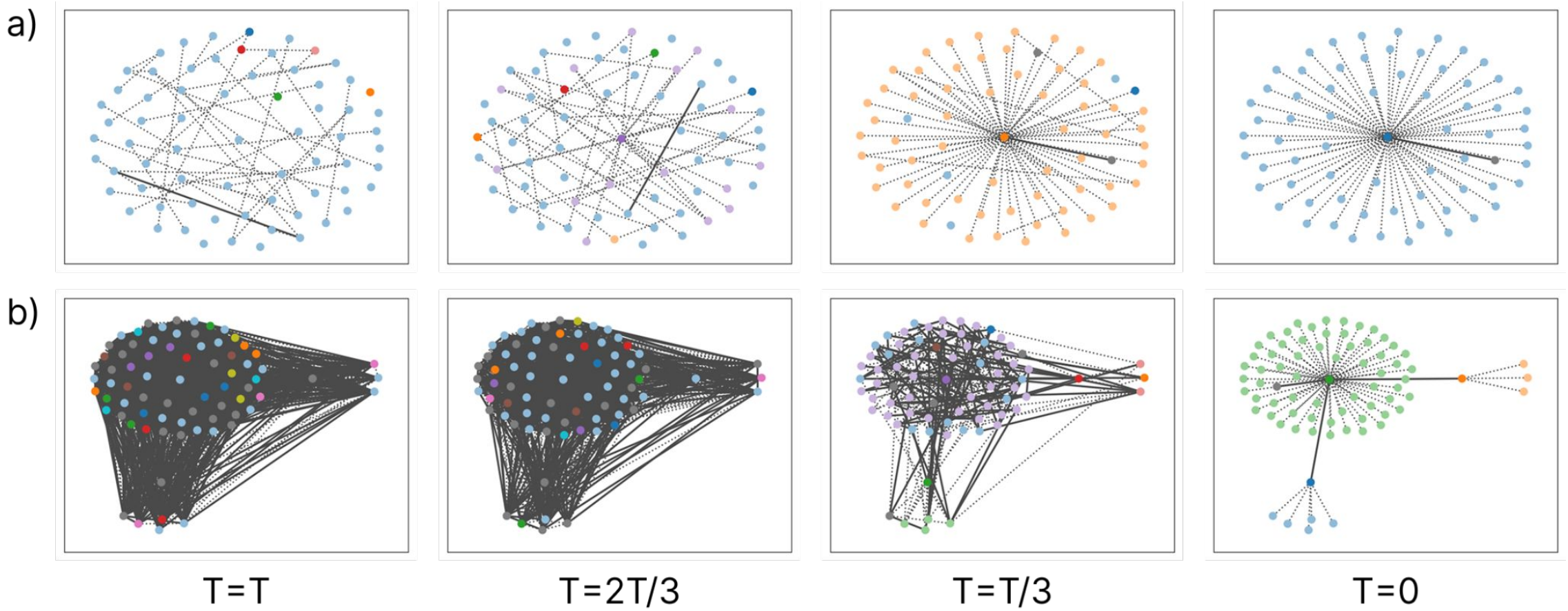

$$E = \begin{array}{c} \\ \text{Root} \\ \text{Clone A} \\ \text{Clone B} \\ \text{Mut 1} \\ \text{Mut 2} \\ \text{Mut 3} \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$X = \begin{array}{cccccc} \text{Root} & \text{Clone A} & \text{Clone B} & \text{Mut 1} & \text{Mut 2} & \text{Mut 3} \end{array} \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

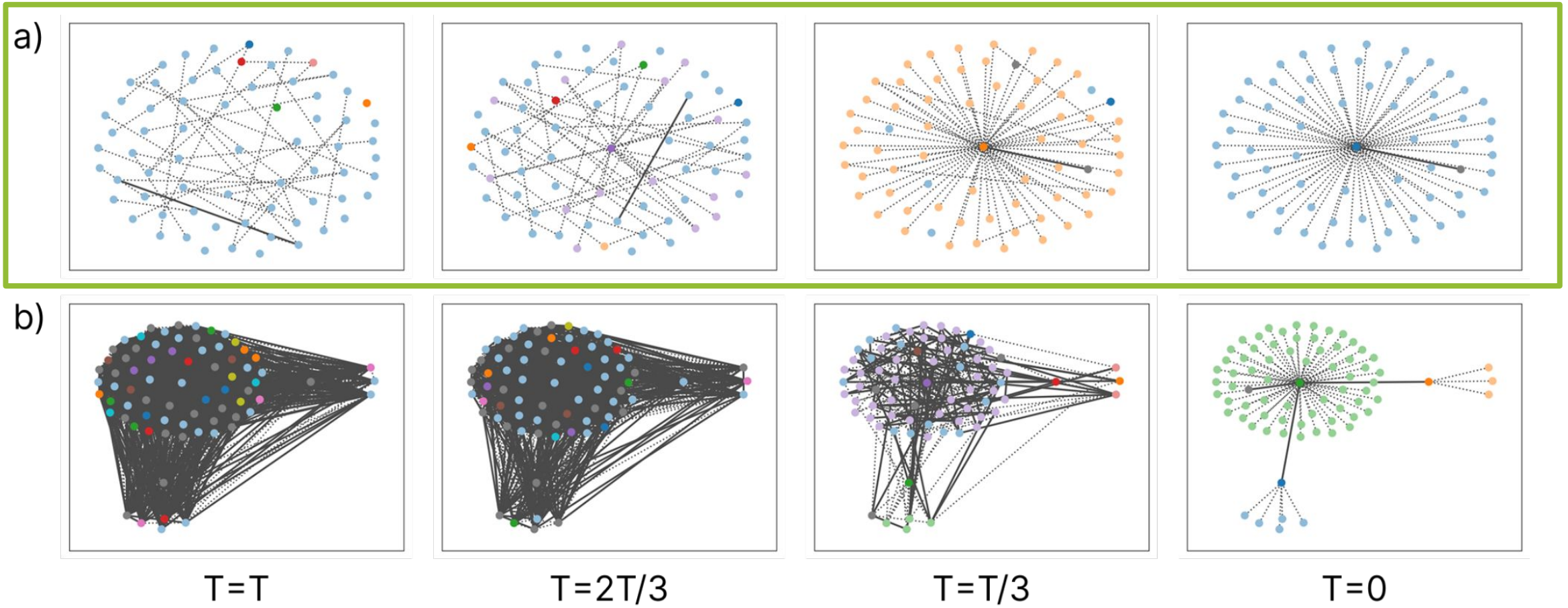
# Training

- Two 7.1 million parameter models trained
  - $\text{DiPhy}_{\text{M}}$: **marginal** noise transition
    - **Preserves class distribution** of nodes and edges during noising process
  - $\text{DiPhy}_{\text{U}}$: **uniform** noise transition
  - All other hyperparameters identical

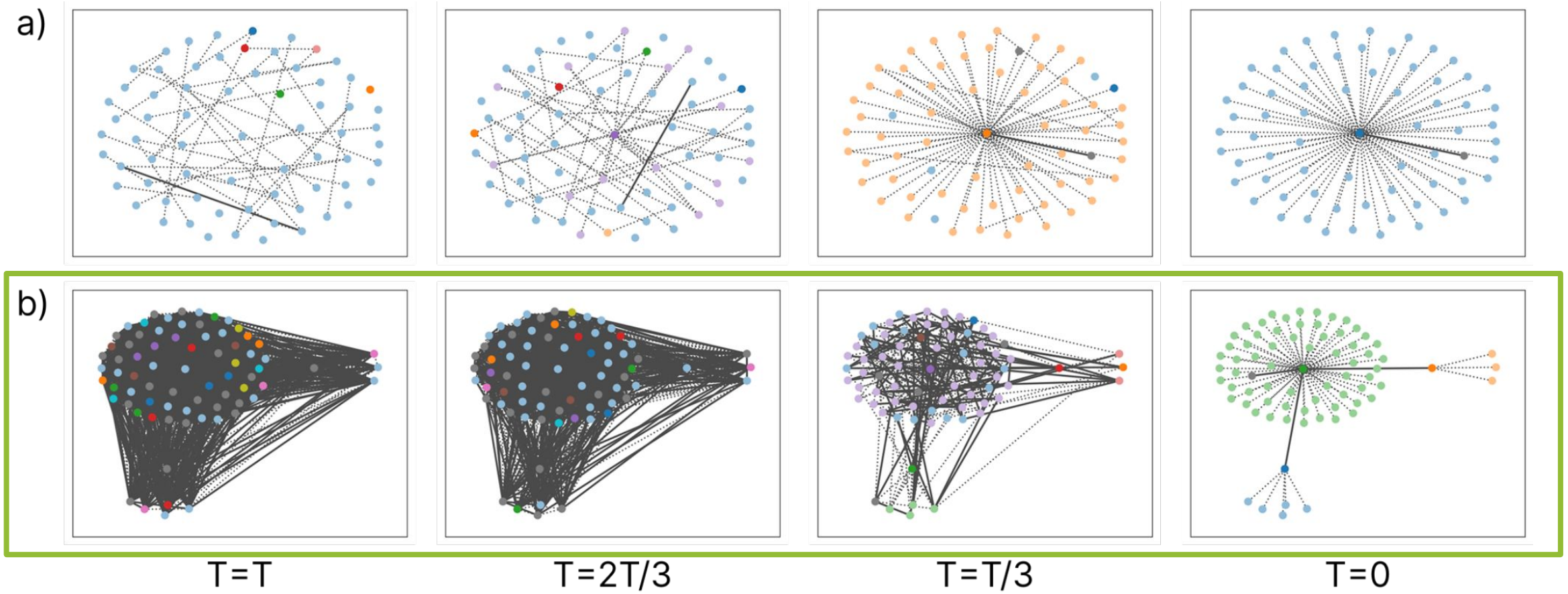
# Marginal vs. Uniform Inference



# Marginal Inference



# Uniform Inference



# Evaluation

- **Compare distributions**
- Node + edge statistics
- Validity test
  - (1) No cycles, (2) exactly one root node with one clone neighbor, (3) clone edges only connect to clones/root, (4) mutation edges only connect mutation nodes to clone nodes

# Outline

Introduction

Methods

**Results**

Discussion

# Generative Performance

| Test Metric | Test Dataset | Uniform | Marginal |
|---|---|---|---|
| num_graphs | 196.0 | 1000.0 | 1000.0 |
| mean_nodes | 80.67 | **80.26** | 78.45 |
| std_nodes | 21.58 | 22.28 | **21.16** |
| mean_edges | 79.67 | **79.37** | 77.48 |
| std_edges | 21.58 | 22.40 | **21.19** |
| mean_clone_fraction | 0.0548 | 0.0332 | **0.0375** |
| mean_mutation_fraction | 0.9321 | 0.9535 | **0.9489** |
| validity_pass_pct | 100.0 | 88.1 | **92.8** |

# Generative Performance

| Test Metric | Test Dataset | Uniform | Marginal |
|---|---|---|---|
| num_graphs | 196.0 | 1000.0 | 1000.0 |
| mean_nodes | 80.67 | **80.26** | 78.45 |
| std_nodes | 21.58 | 22.28 | **21.16** |
| mean_edges | 79.67 | **79.37** | 77.48 |
| std_edges | 21.58 | 22.40 | **21.19** |
| mean_clone_fraction | 0.0548 | 0.0332 | **0.0375** |
| mean_mutation_fraction | 0.9321 | 0.9535 | **0.9489** |
| validity_pass_pct | 100.0 | 88.1 | **92.8** |

# Generative Performance

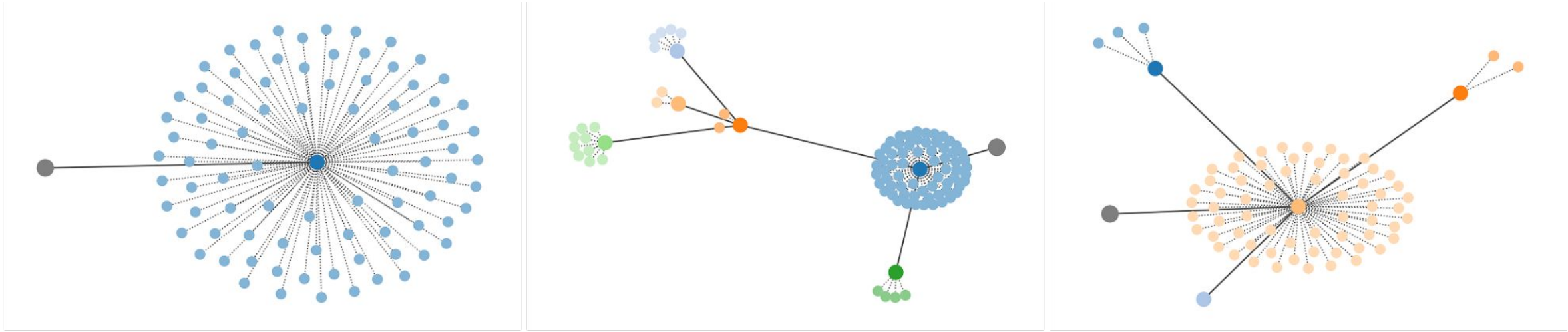
| Test Metric | Test Dataset | Uniform | Marginal |
|---|---|---|---|
| num_graphs | 196.0 | 1000.0 | 1000.0 |
| mean_nodes | 80.67 | **80.26** | 78.45 |
| std_nodes | 21.58 | 22.28 | **21.16** |
| mean_edges | 79.67 | **79.37** | 77.48 |
| std_edges | 21.58 | 22.40 | **21.19** |
| mean_clone_fraction | 0.0548 | 0.0332 | **0.0375** |
| mean_mutation_fraction | 0.9321 | 0.9535 | **0.9489** |
| validity_pass_pct | 100.0 | 88.1 | **92.8** |

# DiPhy$_M$ Performs Best

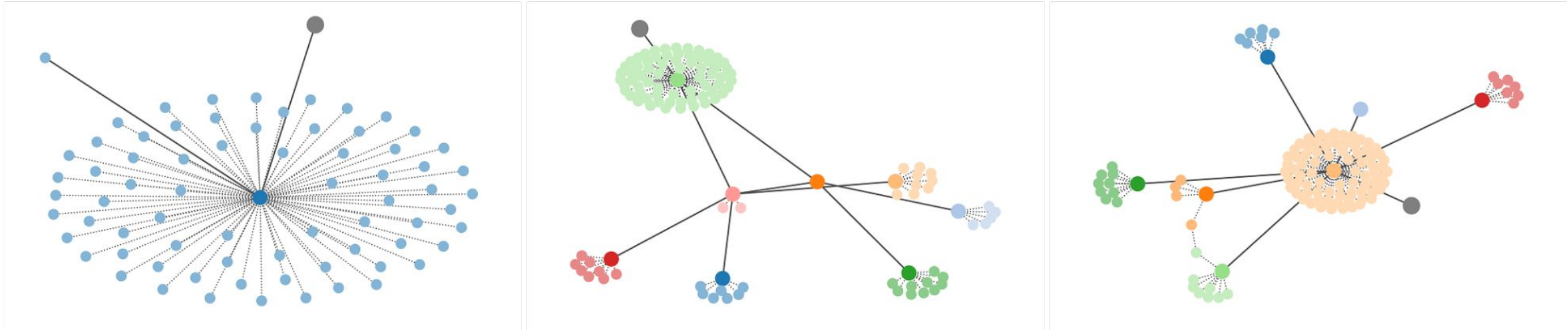
| Test Metric | Test Dataset | Uniform | Marginal |
|---|---|---|---|
| num_graphs | 196.0 | 1000.0 | 1000.0 |
| mean_nodes | 80.67 | **80.26** | 78.45 |
| std_nodes | 21.58 | 22.28 | **21.16** |
| mean_edges | 79.67 | **79.37** | 77.48 |
| std_edges | 21.58 | 22.40 | **21.19** |
| mean_clone_fraction | 0.0548 | 0.0332 | **0.0375** |
| mean_mutation_fraction | 0.9321 | 0.9535 | **0.9489** |
| validity_pass_pct | 100.0 | 88.1 | **92.8** |

# Example Graphs

a)

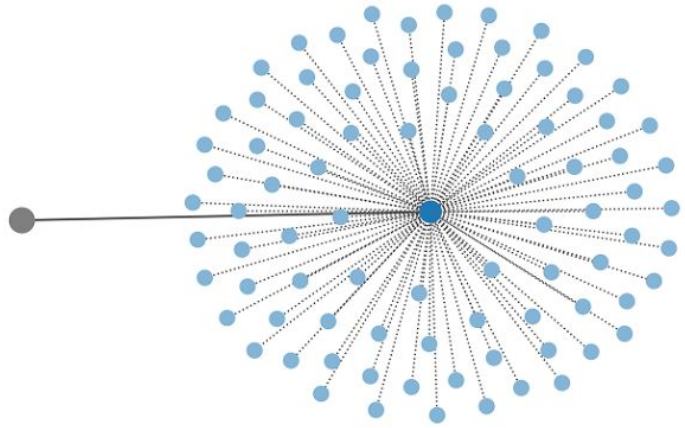


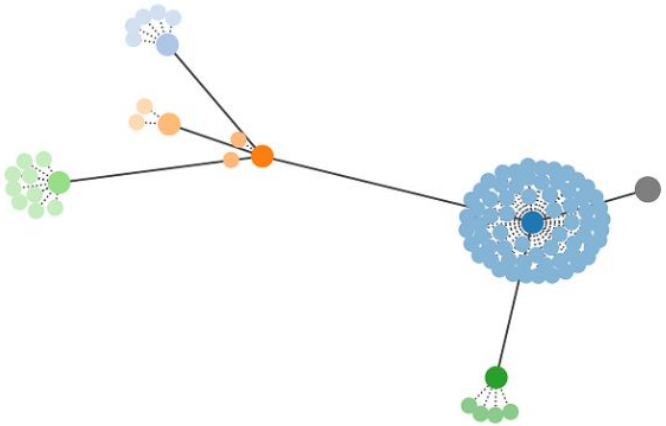
b)

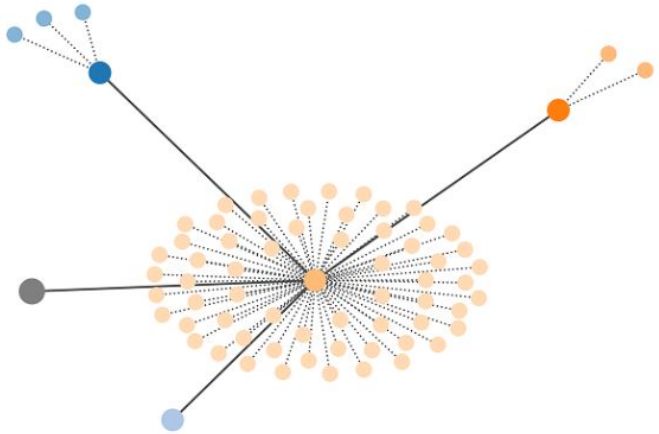


# Example Graphs

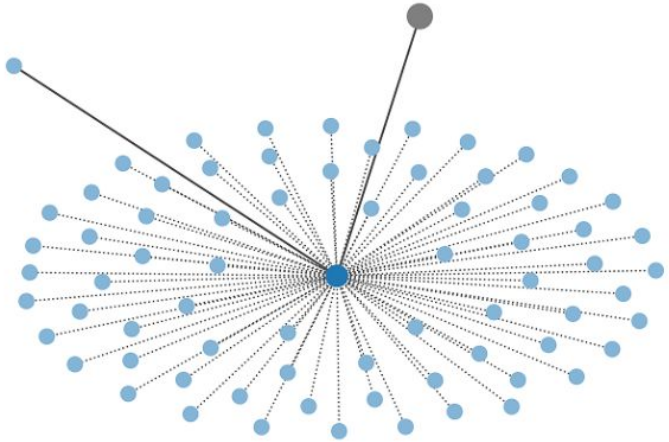
a)

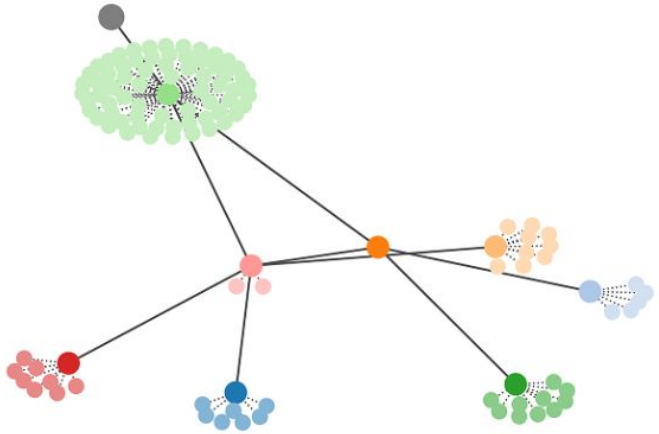


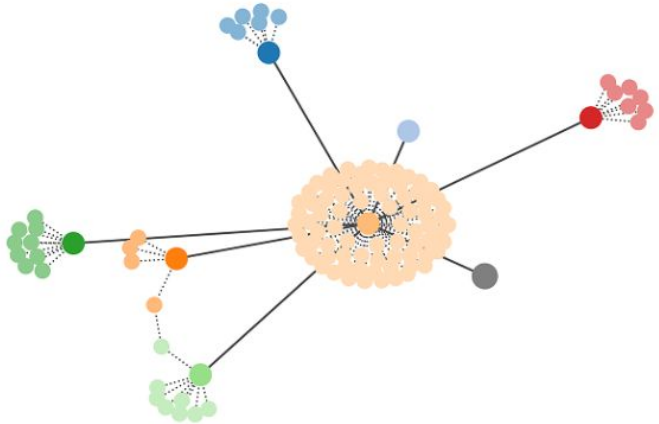




b)

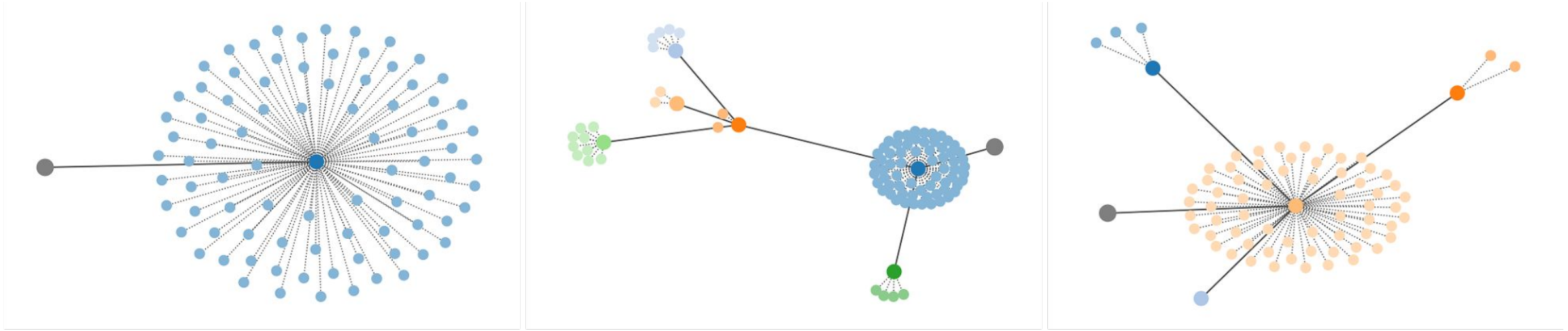




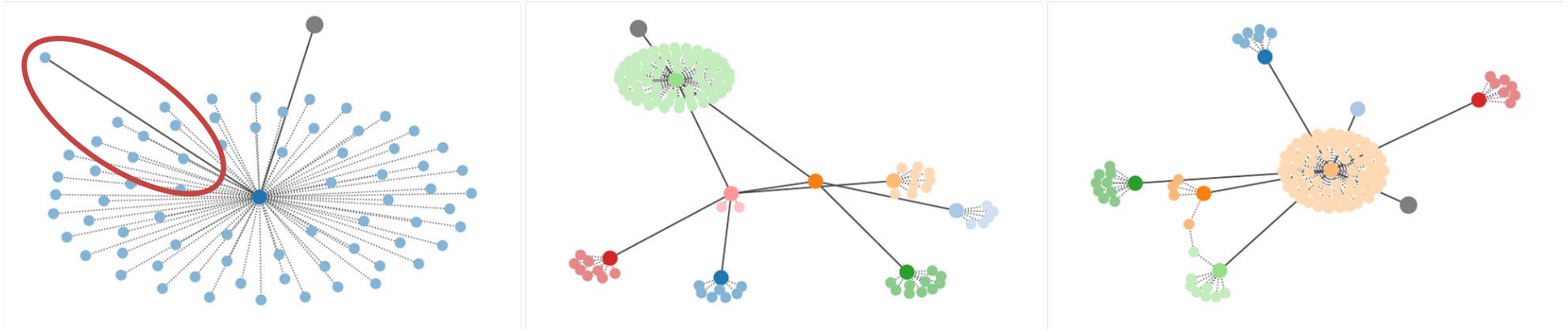


# Invalid Edges

a)

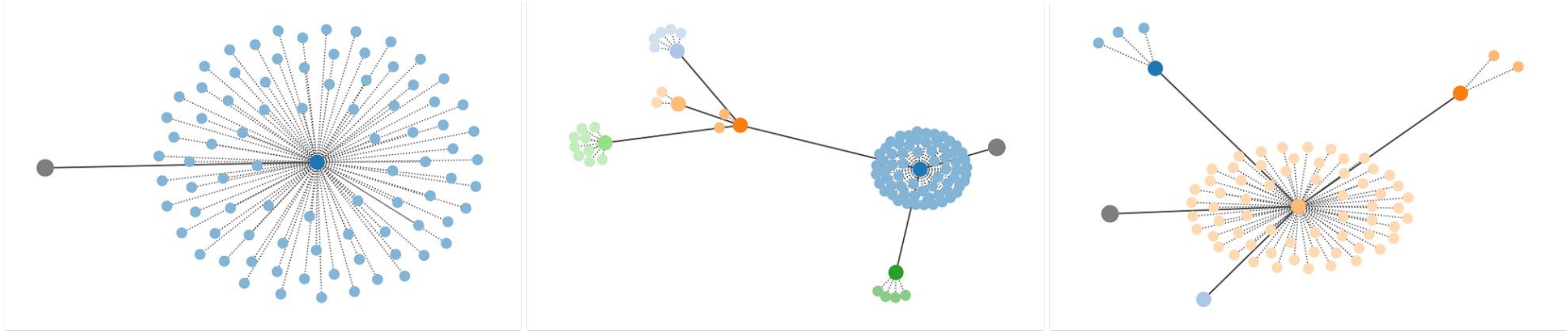


b)

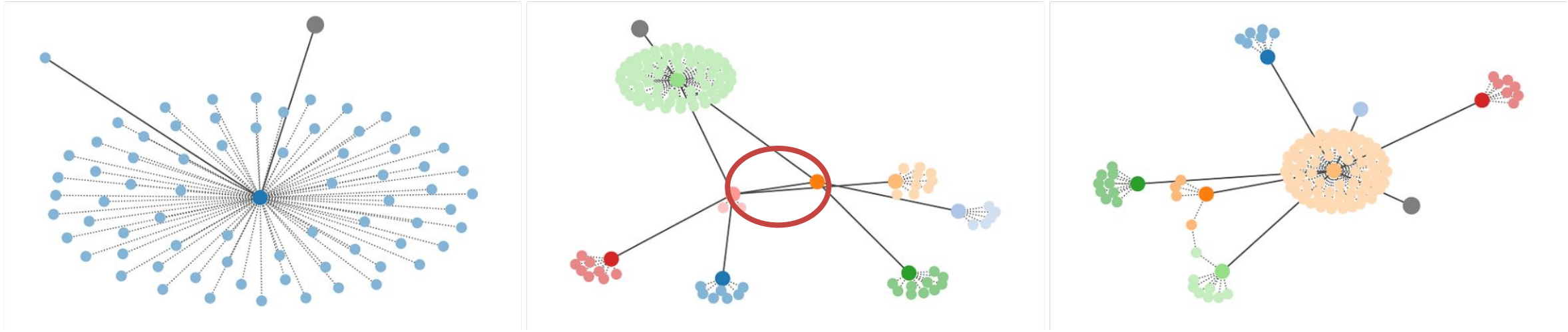


# Invalid Edges

a)

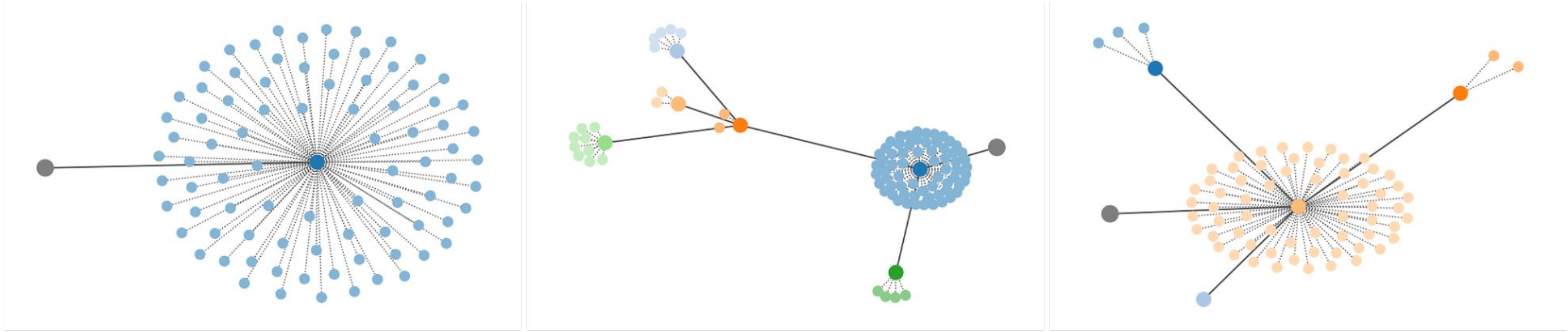


b)

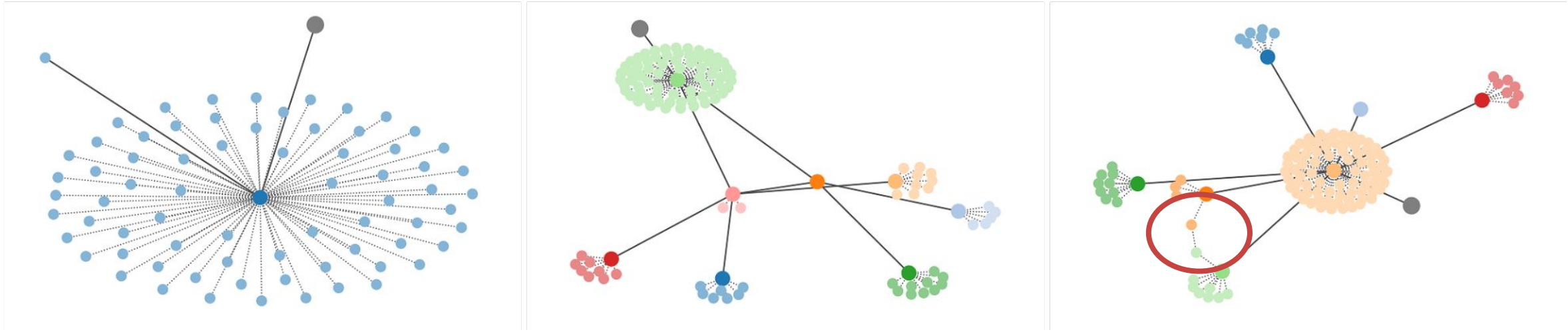


# Invalid Edges

a)



b)



# Inference Speed

- Cannot directly compare with traditional methods
- 1,000 graphs, 100 visualized, 20 animated
- **~4 hours** total with a single NVIDIA 32GB V100 GPU

# Outline

Introduction

Methods

Results

**Discussion**

# Discussion

# Discussion

More data, bigger model

# Discussion

More data, bigger model

Guided sampling

# Discussion

More data, bigger model

Guided sampling

Custom architecture

# Discussion

More data, bigger model

Guided sampling

Custom architecture

Use cases

# Conclusion

- Deep **generative approach** for phylogeny reconstruction
- Developed **synthetic training** dataset and **encoding**
- Showed 7.1 million parameter diffusion model can **learn phylogeny structures**
- Initial, proof of concept

# Acknowledgements