GRAPH GANS FOR HIGH ENERGY PHYSICS DATA GENERATION

Raghav Kansal, *Javier Duarte, Breno Orzari, Thiago Tomei, Maurizio Pierini, Mary Touranakou, Jean-Roch Vlimant, Dimitrios Gunopulos*

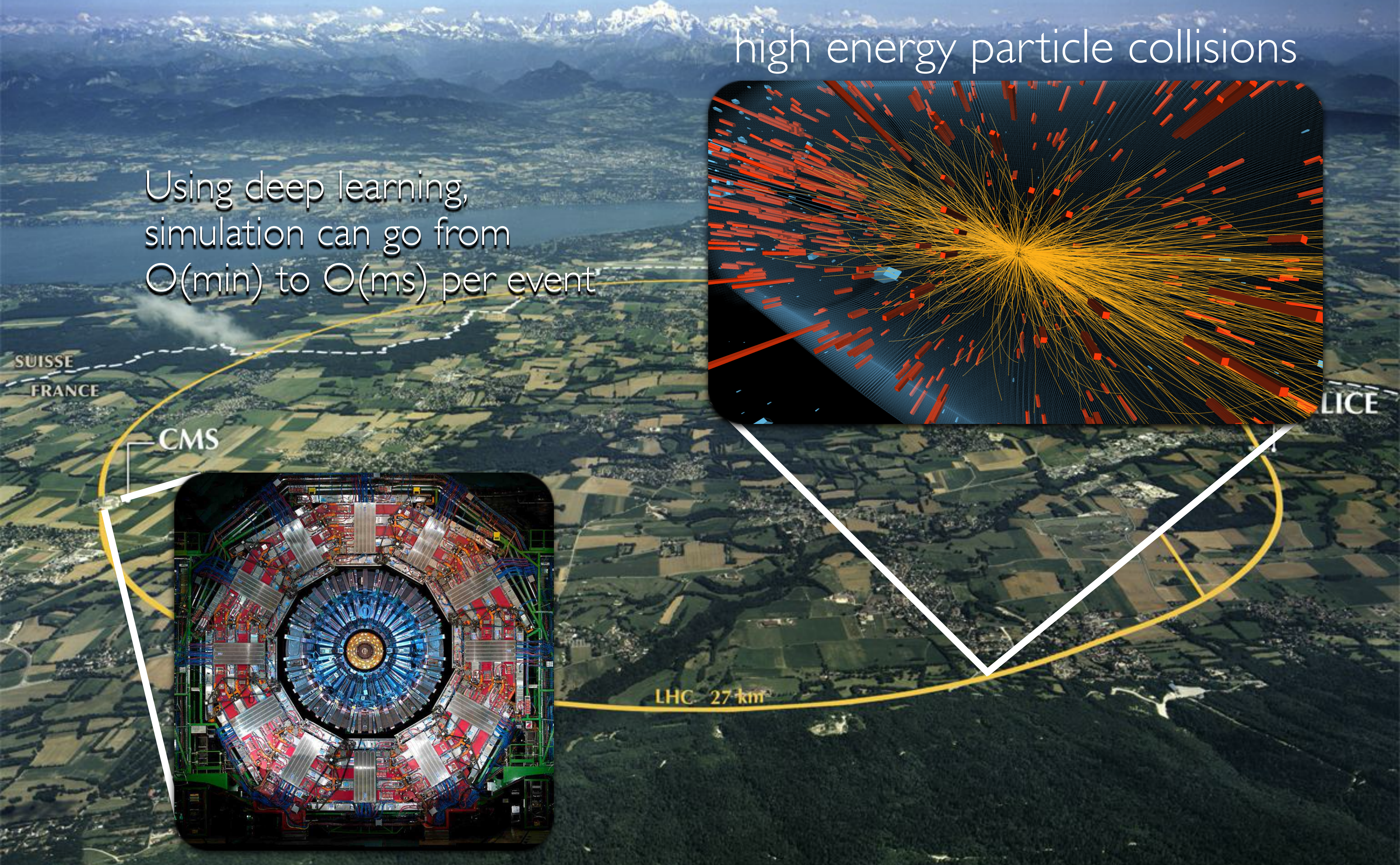
Berkeley Deep Generative Models for Fundamental Physics Meeting

17/3/21

LARGE HADRON COLLIDER

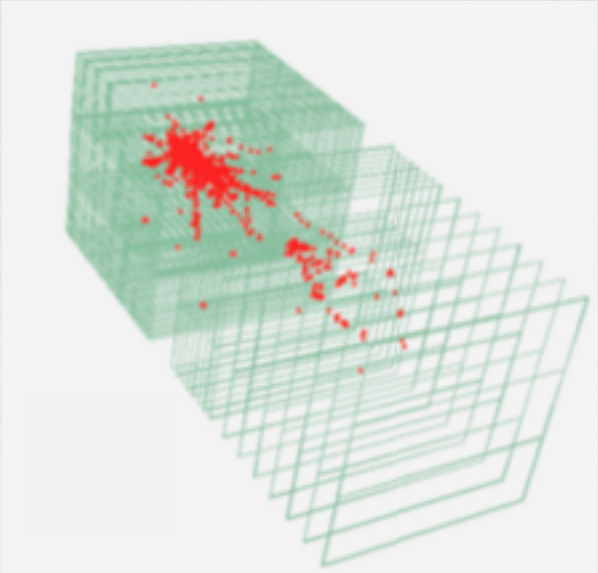
high energy particle collisions

Using deep learning, simulation can go from O(min) to O(ms) per event



WHY GRAPHS?

- Properties of HEP data:
  - High granularity
  - Sparsity
  - Irregular geometry



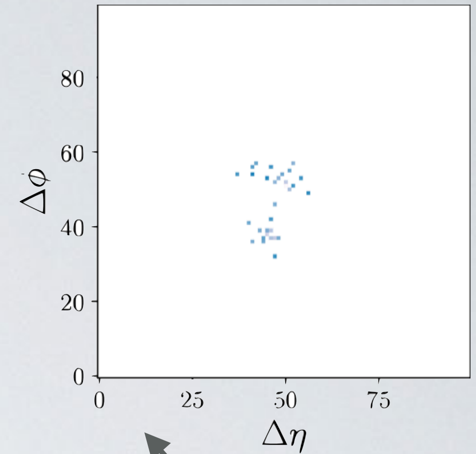
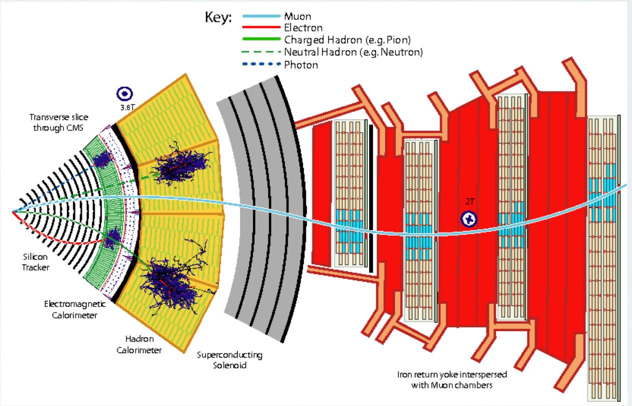
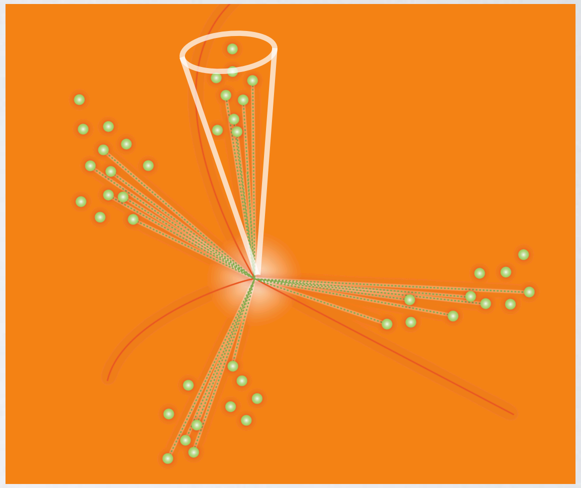


Image repr. for CNNs

Jets

Raw/Reconstructed Detector Data





RELATED WORK

LAGAN (Oliveira et al. 2017)

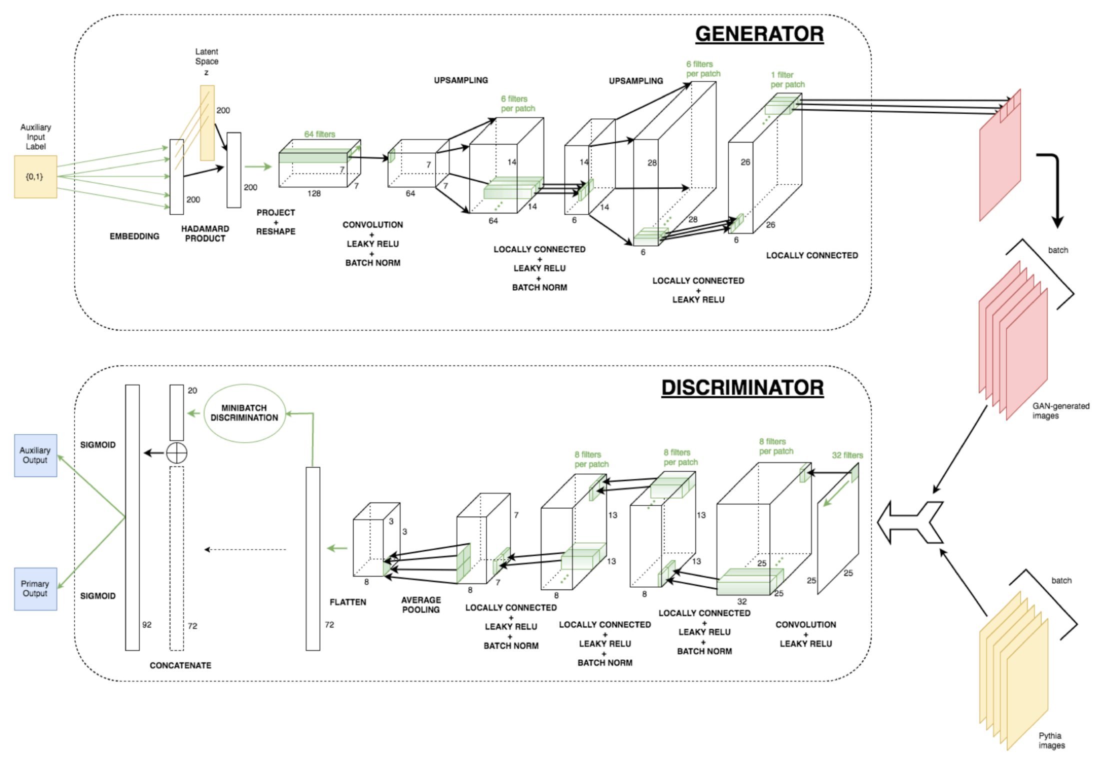
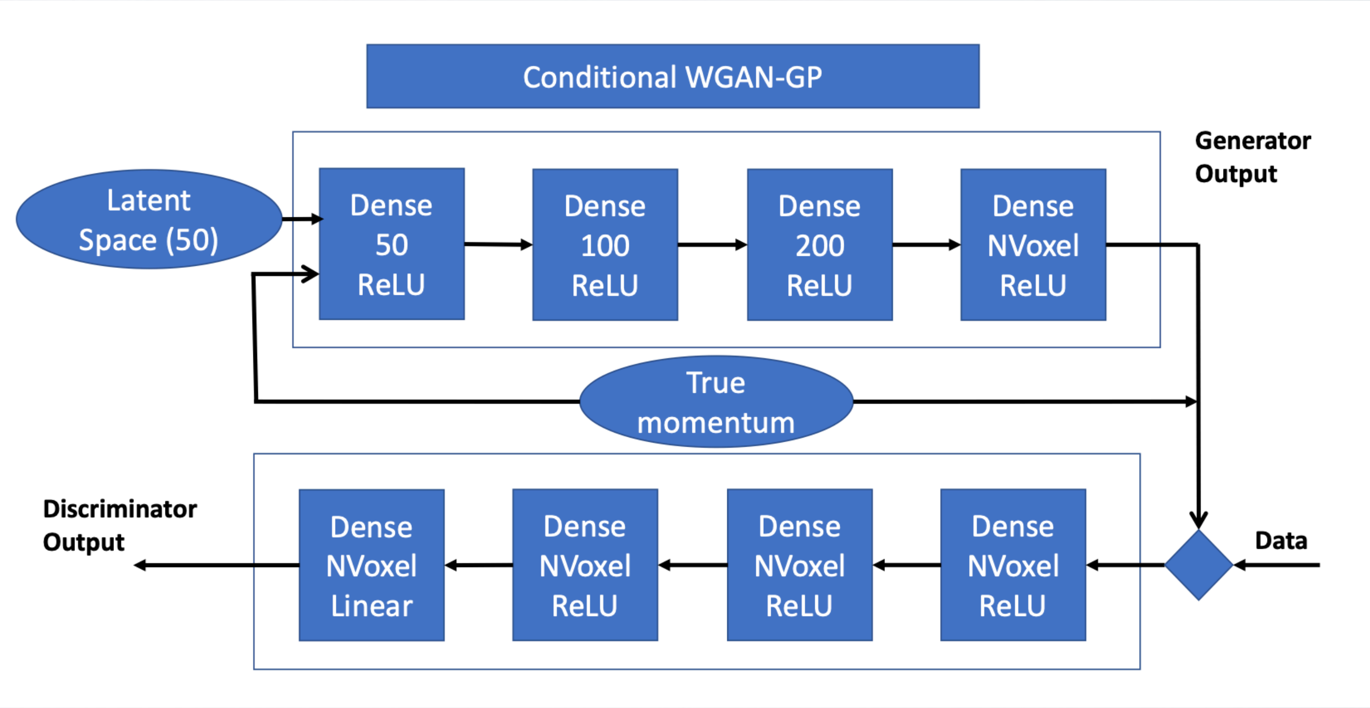
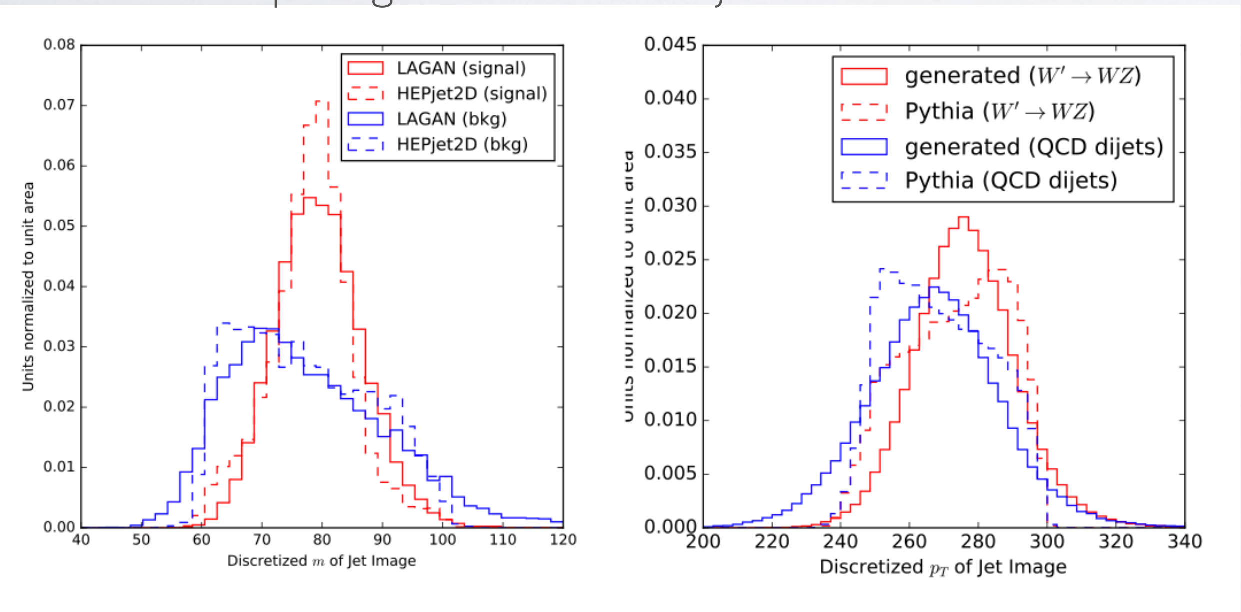


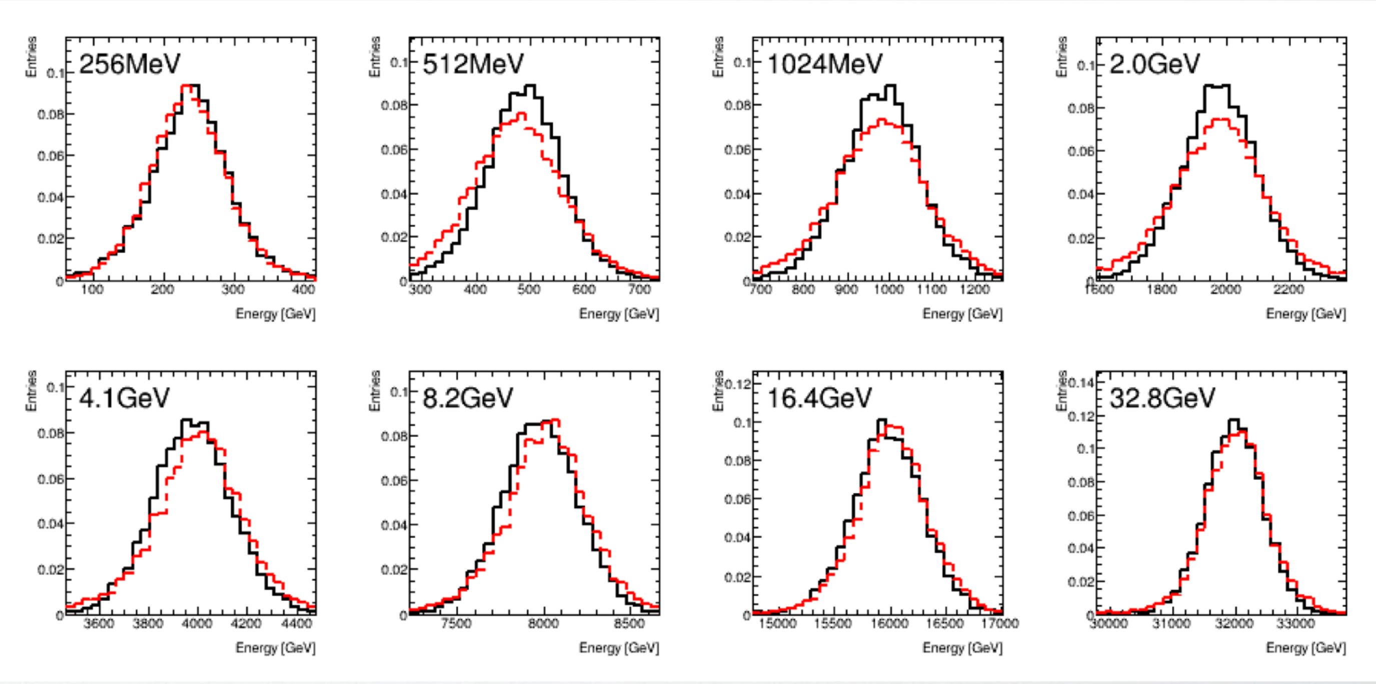
Figure 4: LAGAN architecture

FastCaloGAN (ATLAS Collab. 2020)



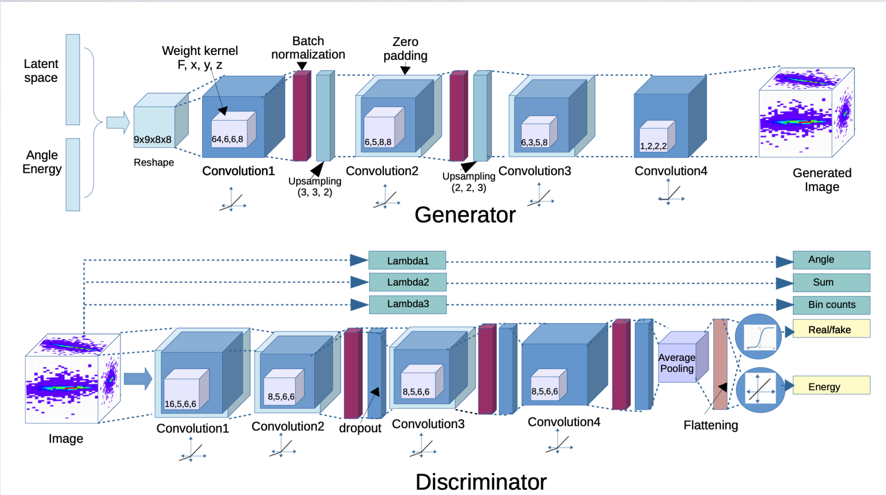
Comparing distributions of jet features

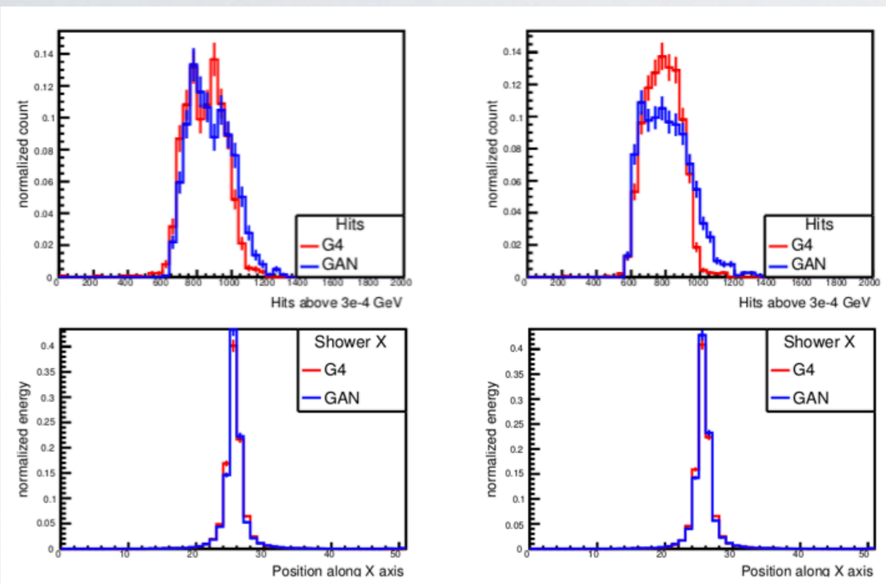




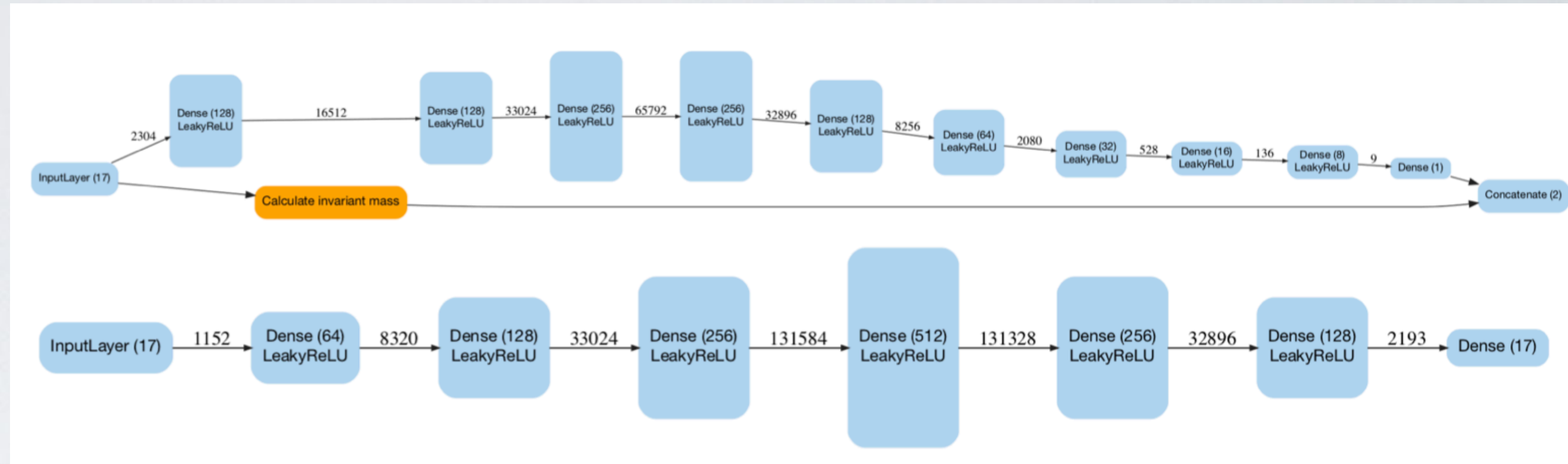
RELATED WORK

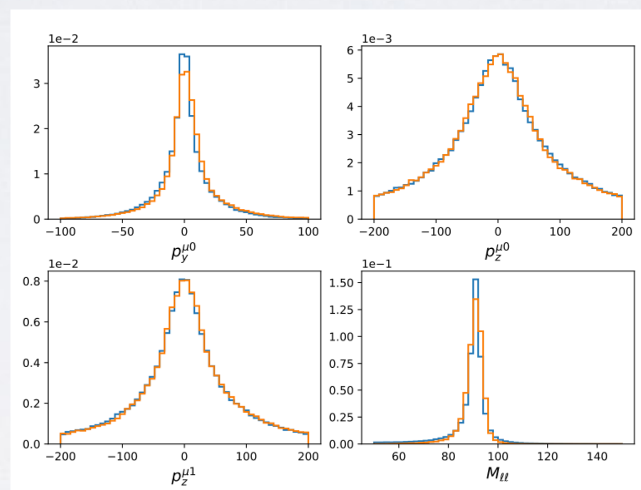
DL for Calorimetry (Belayneh et al. 2020)



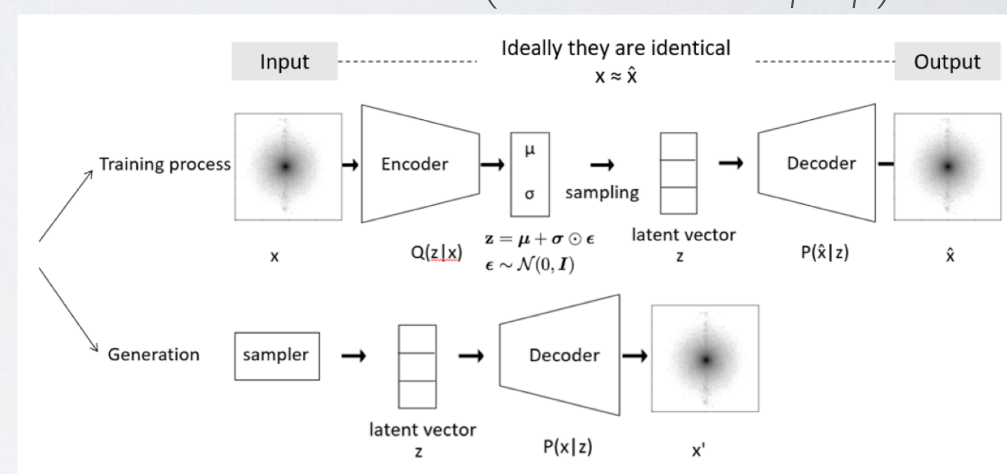


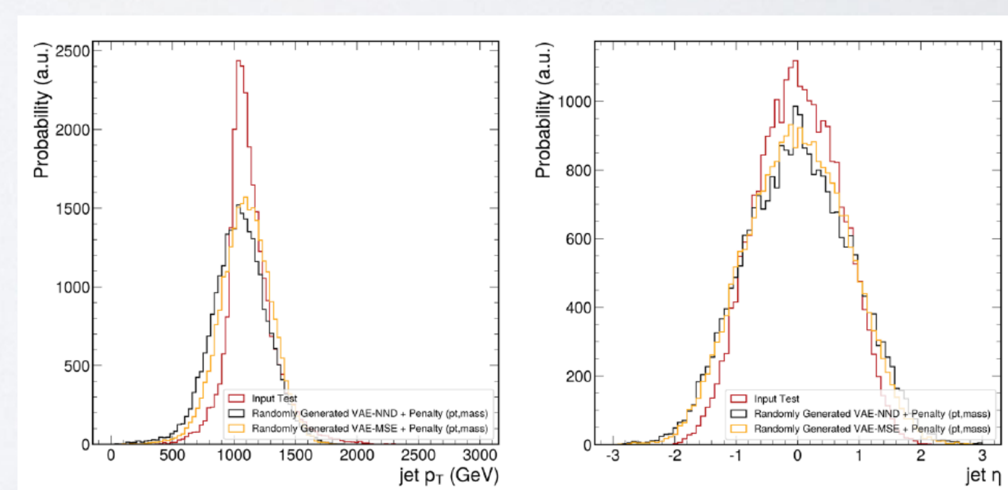
Hashemi et al. 2019



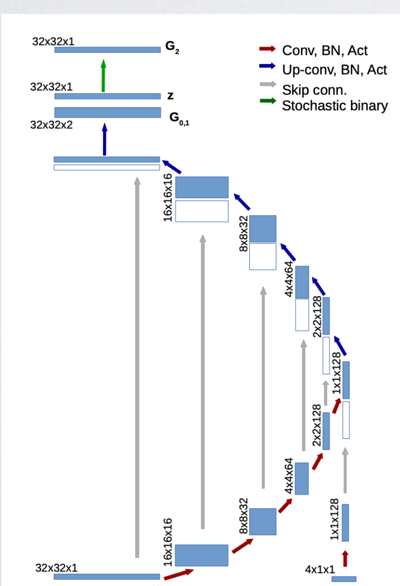


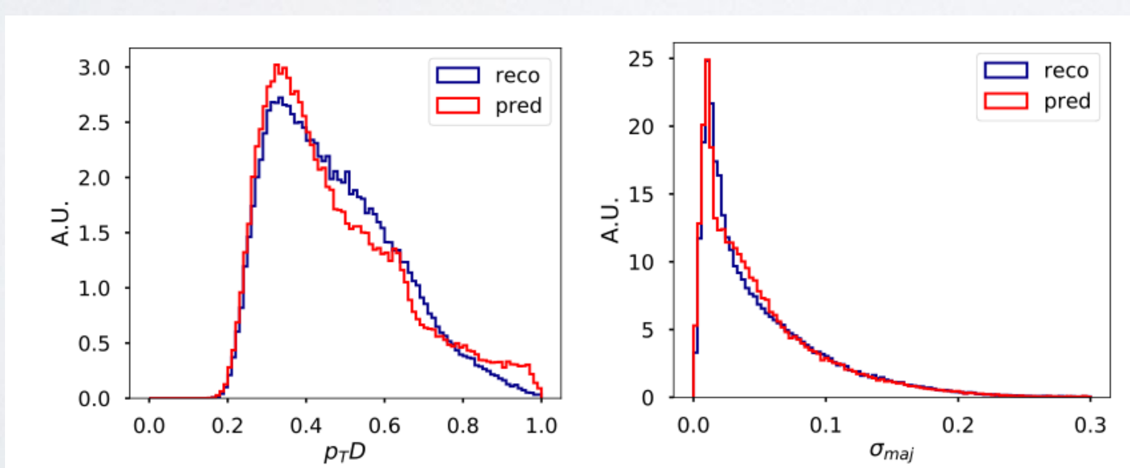
VAE for FastSim (Orzari et al. *in prep*)





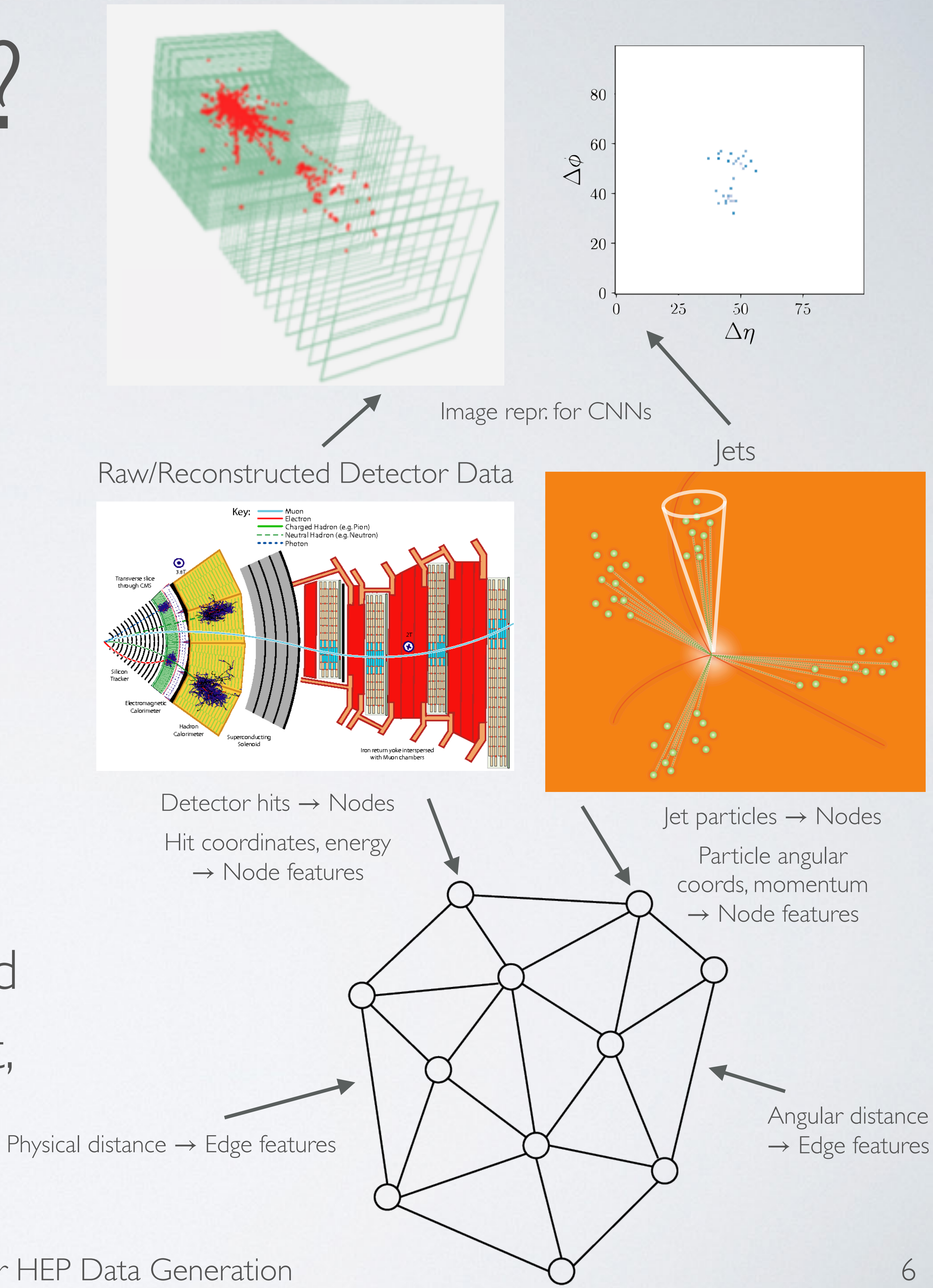
Musella et al. 2018





WHY GRAPHS?

- Properties of HEP data:
  - High granularity
  - Sparsity
  - Irregular geometry
- ⇒ Graphs furnish a more natural representation of such data than images
- Our goal is to design a graph-based GAN, which may be more efficient, flexible, and better suited to data

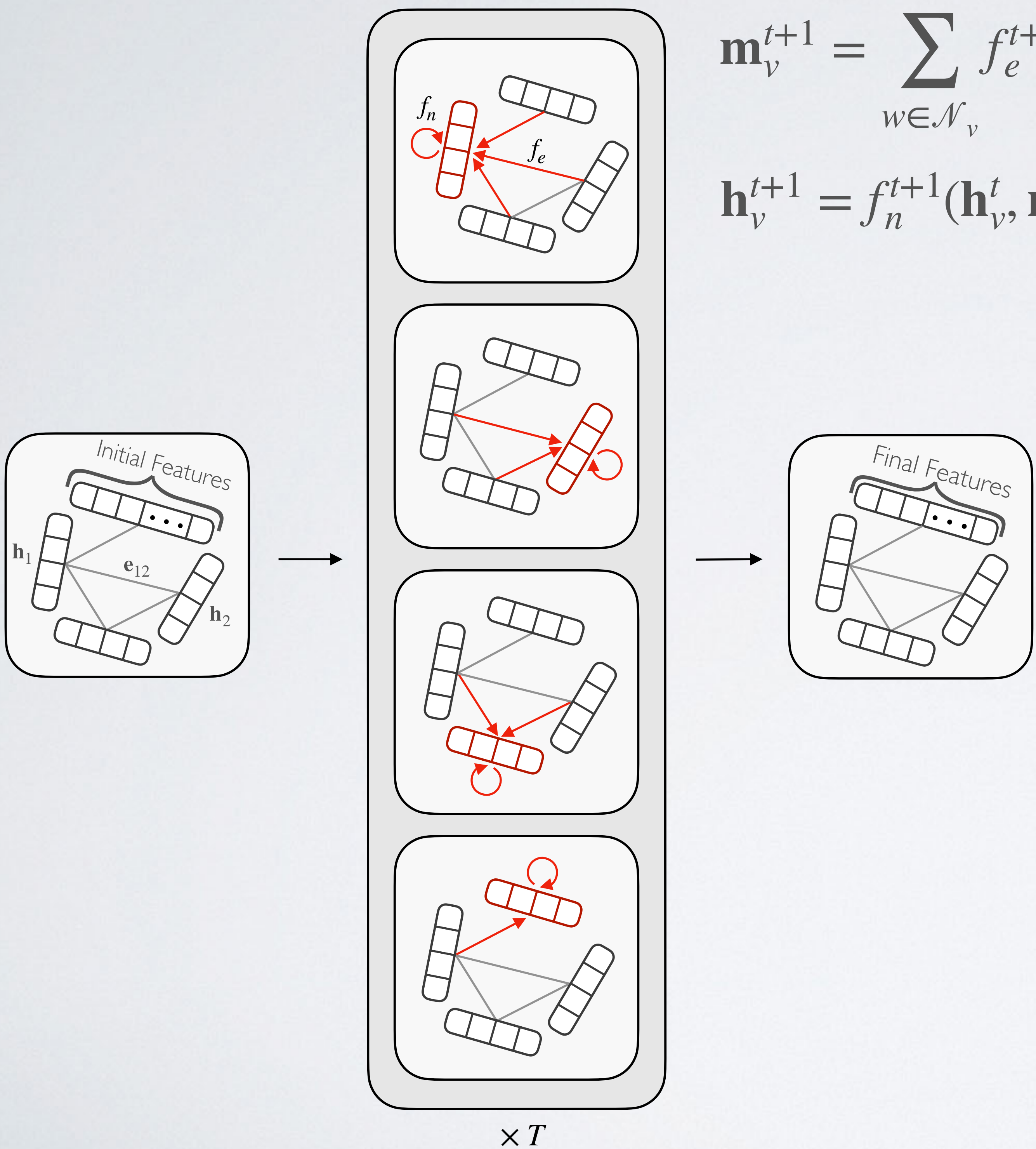


GAN ARCHITECTURE

Nodes in the graph learn via 'message passing' between their neighbours:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1})$$

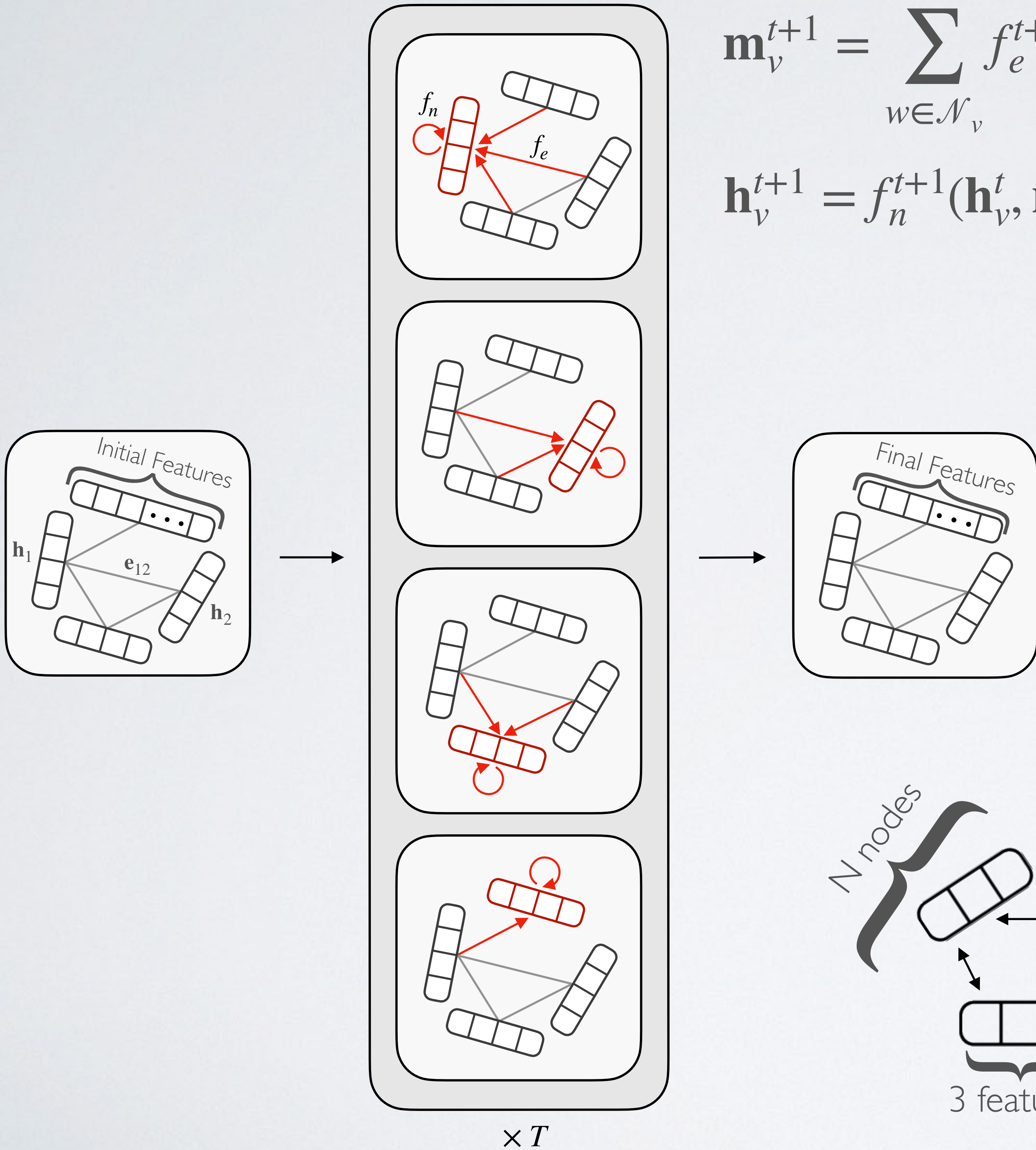


GAN ARCHITECTURE

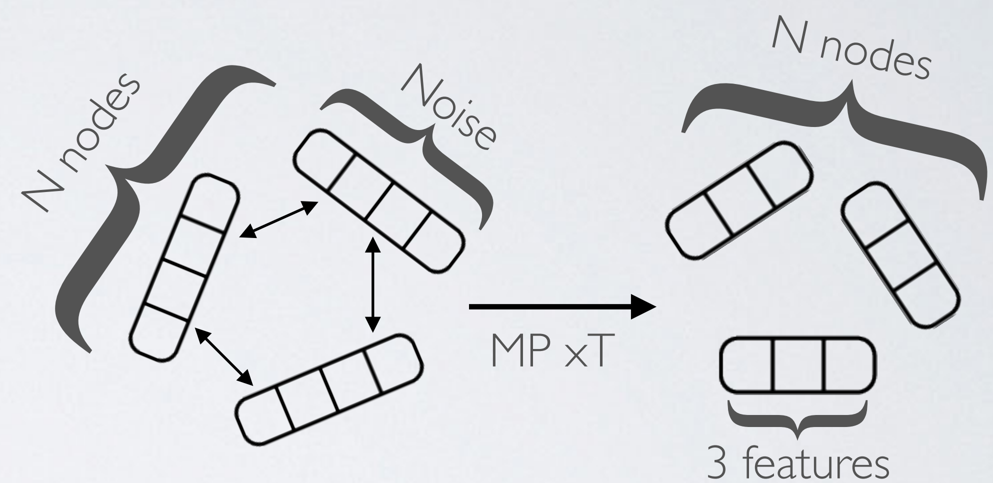
Such a message passing model is used in both the generator and discriminator:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t)$$

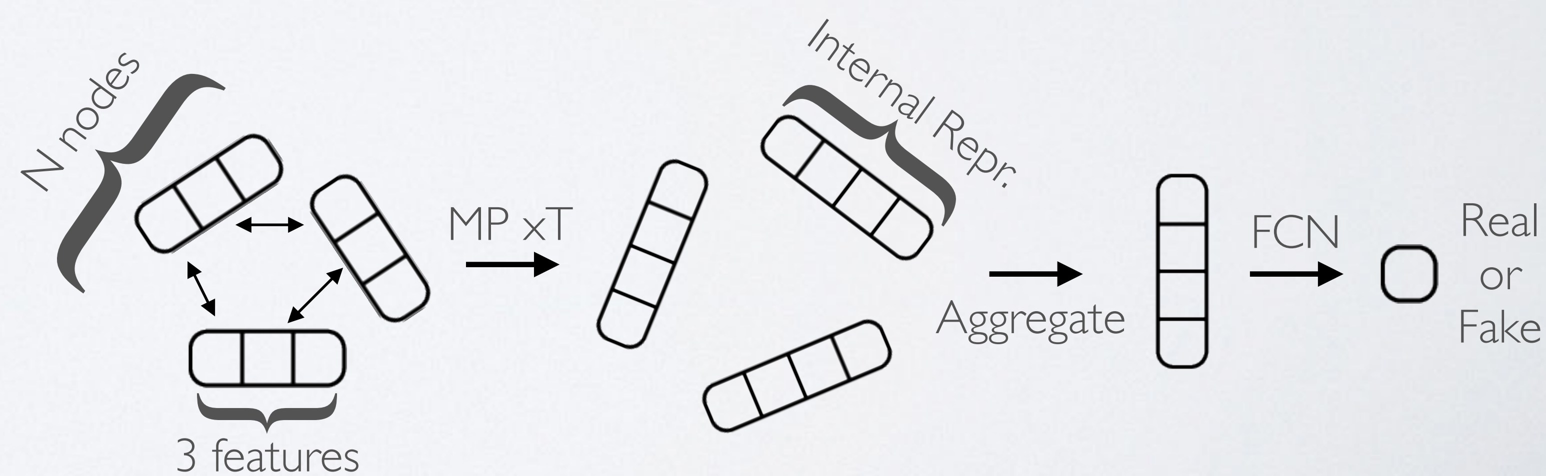
$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1})$$



Generator



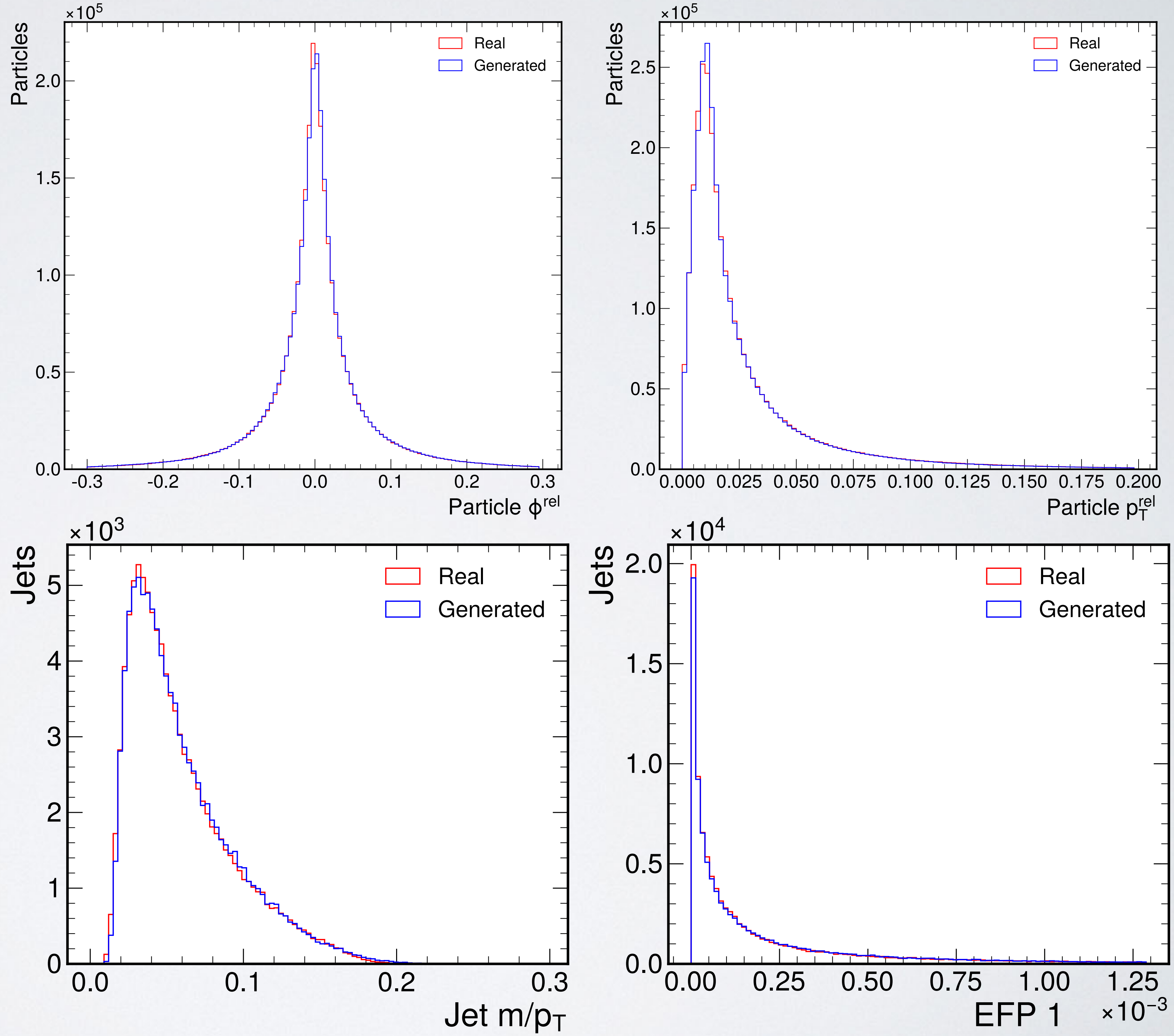
Discriminator



RESULTS

Up to 30 particles (zero-padding when needed) per **gluon** jets of $p_T \sim 1\,\text{TeV}$

- Real distributions are reproduced with high fidelity:
- We look at jet mass and energy flow polynomials (EFPs) (Komiske et al. 2017) - high-order particle correlations which form a basis for all useful jet observables
- Remarkable that complex jet features are learnt without any specific guidance

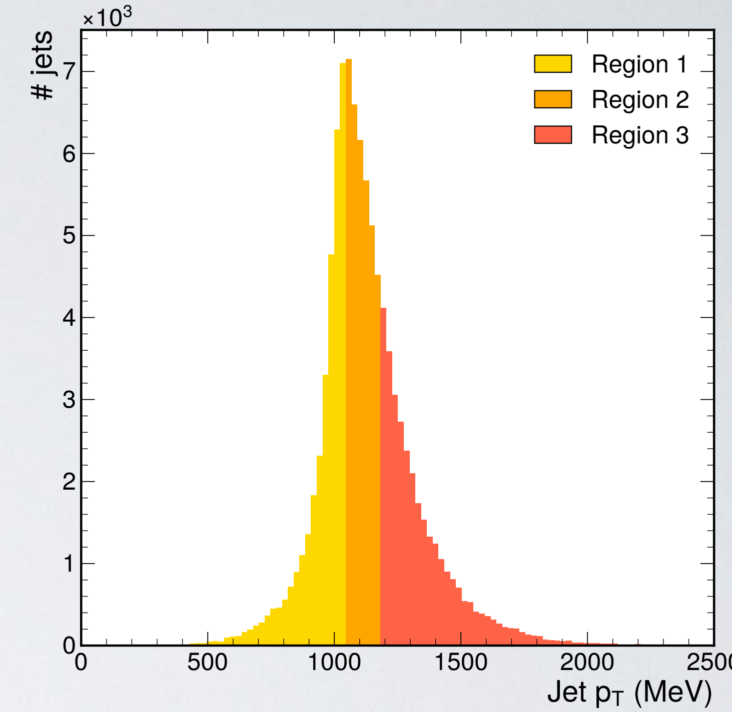


CONDITIONAL GAN

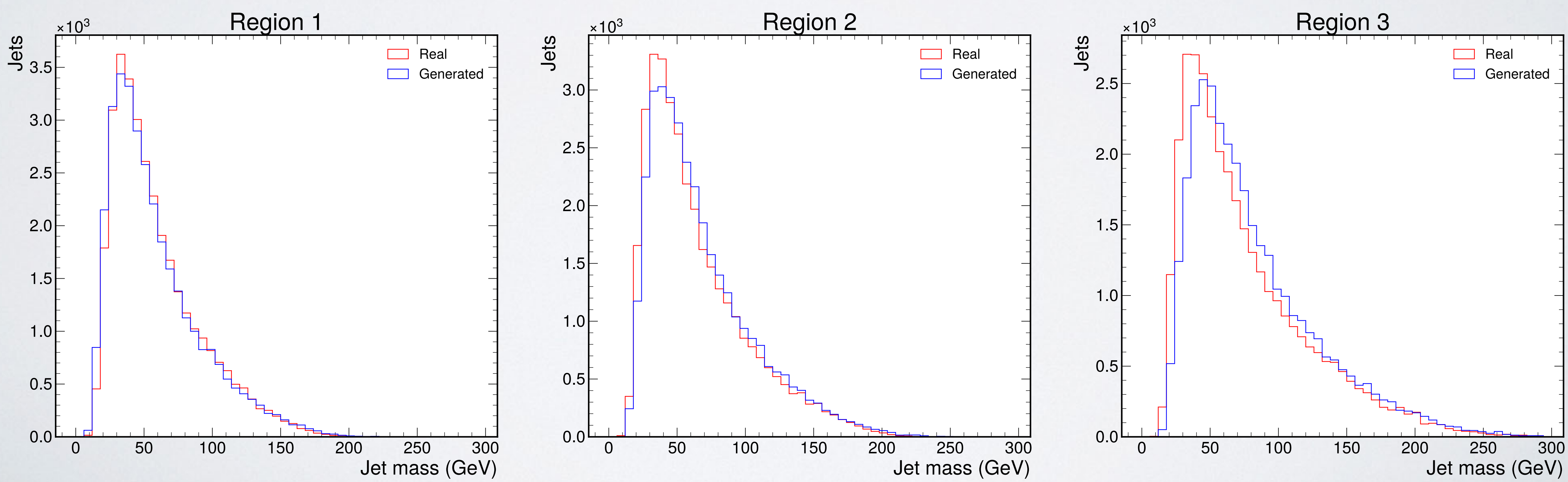
- We want to be able to condition the generator on jet p_T, η

- We modified the message passing architecture to incorporate graph-level features, and trained a GAN conditioned on p_T

- For evaluation the jets have been split into three regions with ~33k jets each



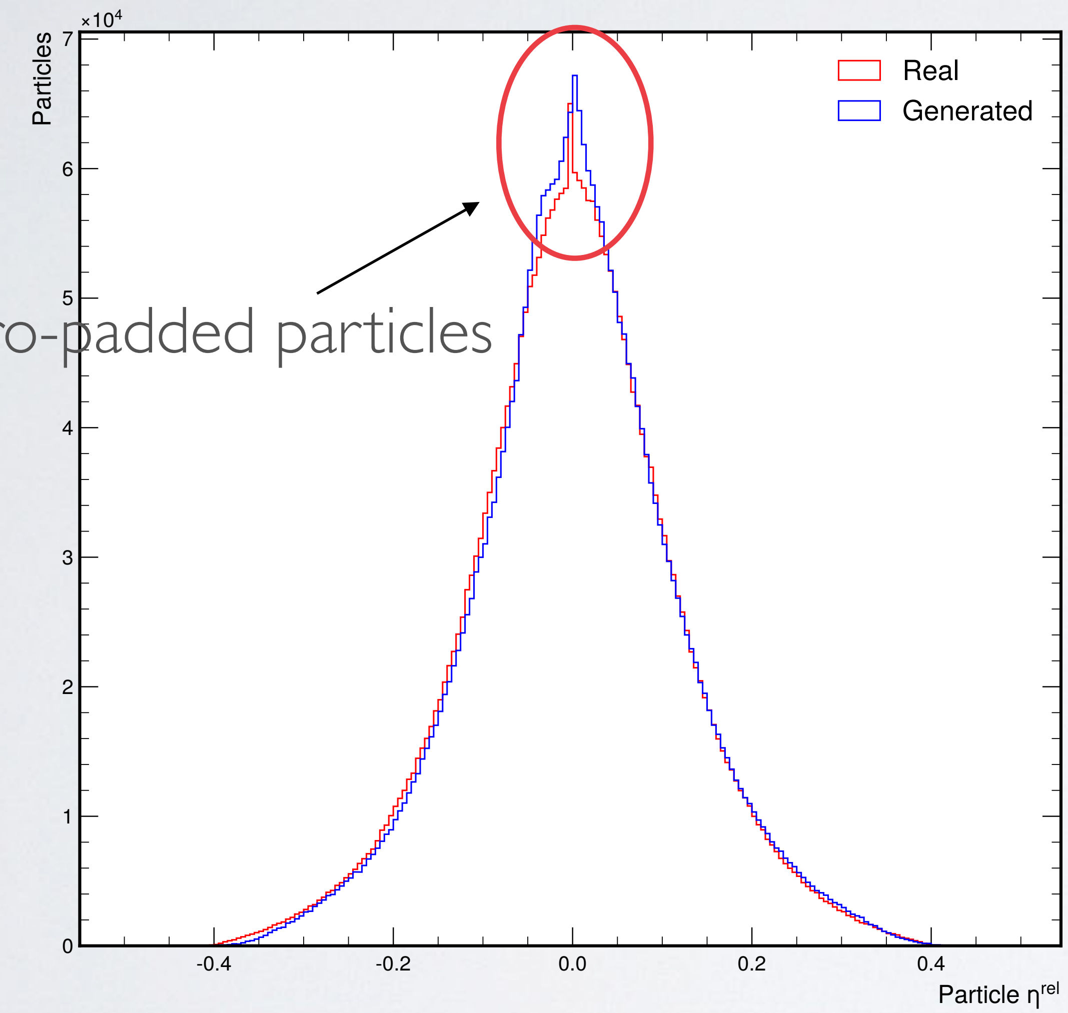
- Distributions well-matched in all three regions, but some degradation for higher p_T:



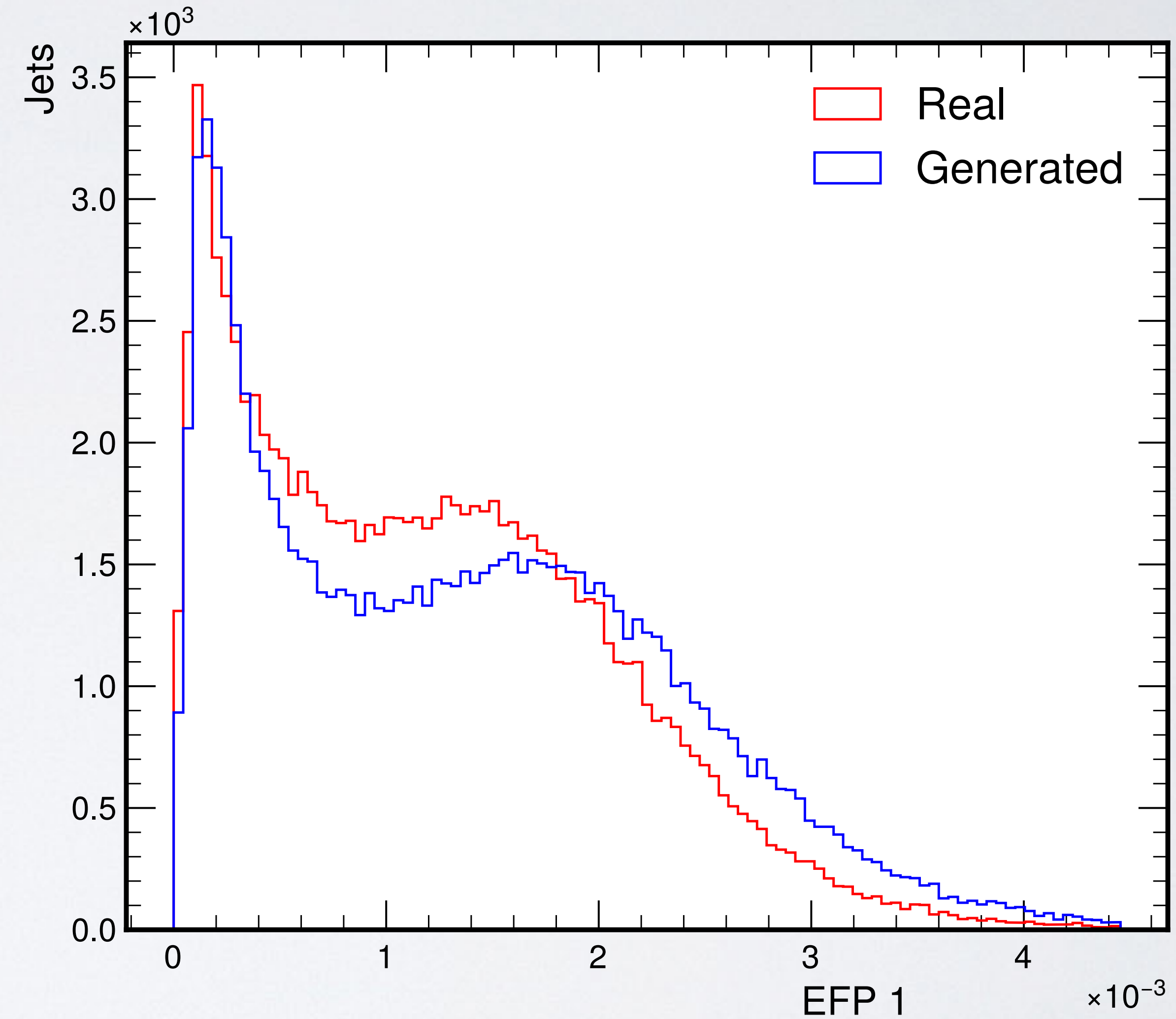
VARIABLE-SIZED GRAPHS

Up to 30 particles (zero-padding when needed) per **top** jets of $p_T \sim 1$ TeV

- Some issues with generation because of zero-padded particles:



Zero-padded particles



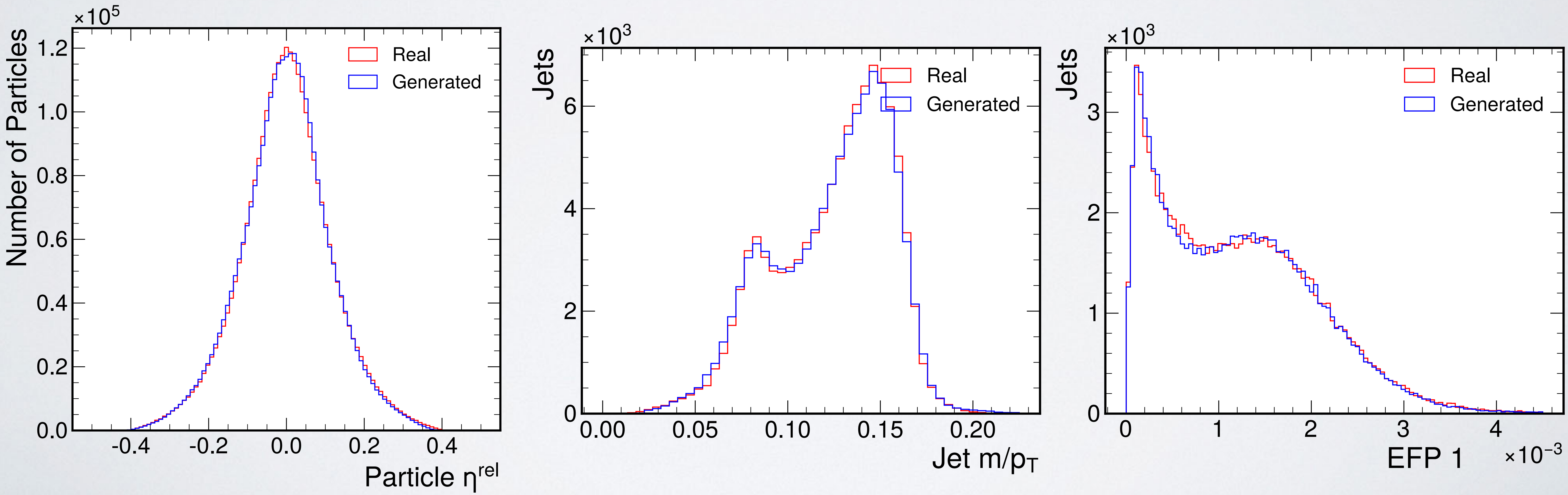
- (Also top jets are more complicated: $t \to Wb \to qqb$ so 2 or 3 subjets)

VARIABLE-SIZED GRAPHS

Up to 30 particles per **top** jets of $p_T \sim 1$ TeV

- We 'mask' the particles with an extra binary feature, telling the model if they're real or zero-padded - effectively allowing for variable sized graphs

- With our masking strategy, distributions are well-matched even for complex top jets:



SUMMARY

Paper: https://arxiv.org/abs/2012.00173
Code: https://github.com/rkansal47/graph-gan
Email: rkansal@ucsd.edu

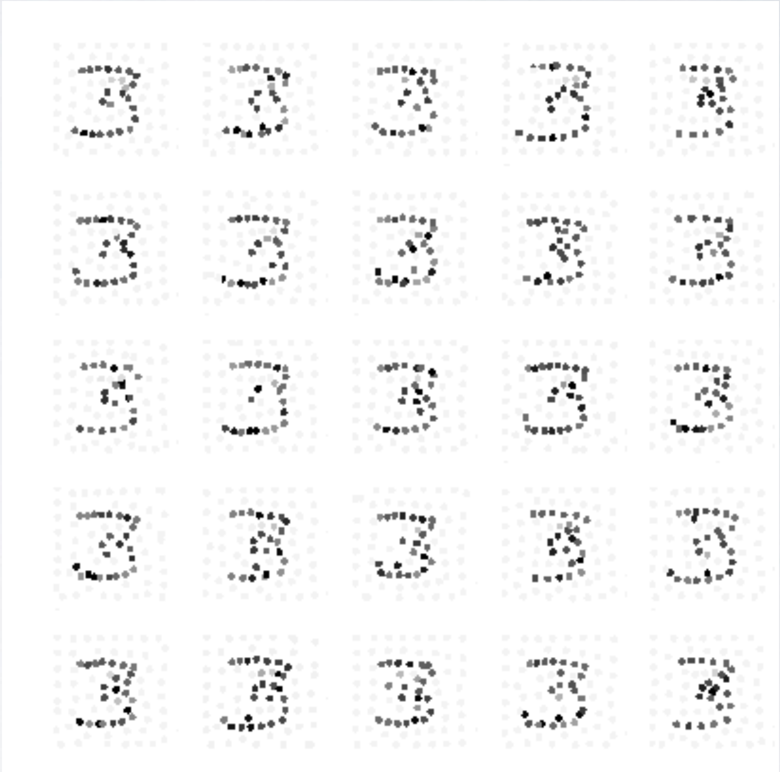
- We developed a new graph-based GAN, which may be naturally suited for HEP data
- It has been successful in producing jets (as well as graphical MNIST data)
- Currently exploring conditioning and variable-sized graphs
- Future work
  - Further applications to HEP datasets (calorimeter data) and beyond
  - More sophisticated architectures e.g. sequential generation, Lorentz group equivariant networks

BACKUP

GAN TRAINING

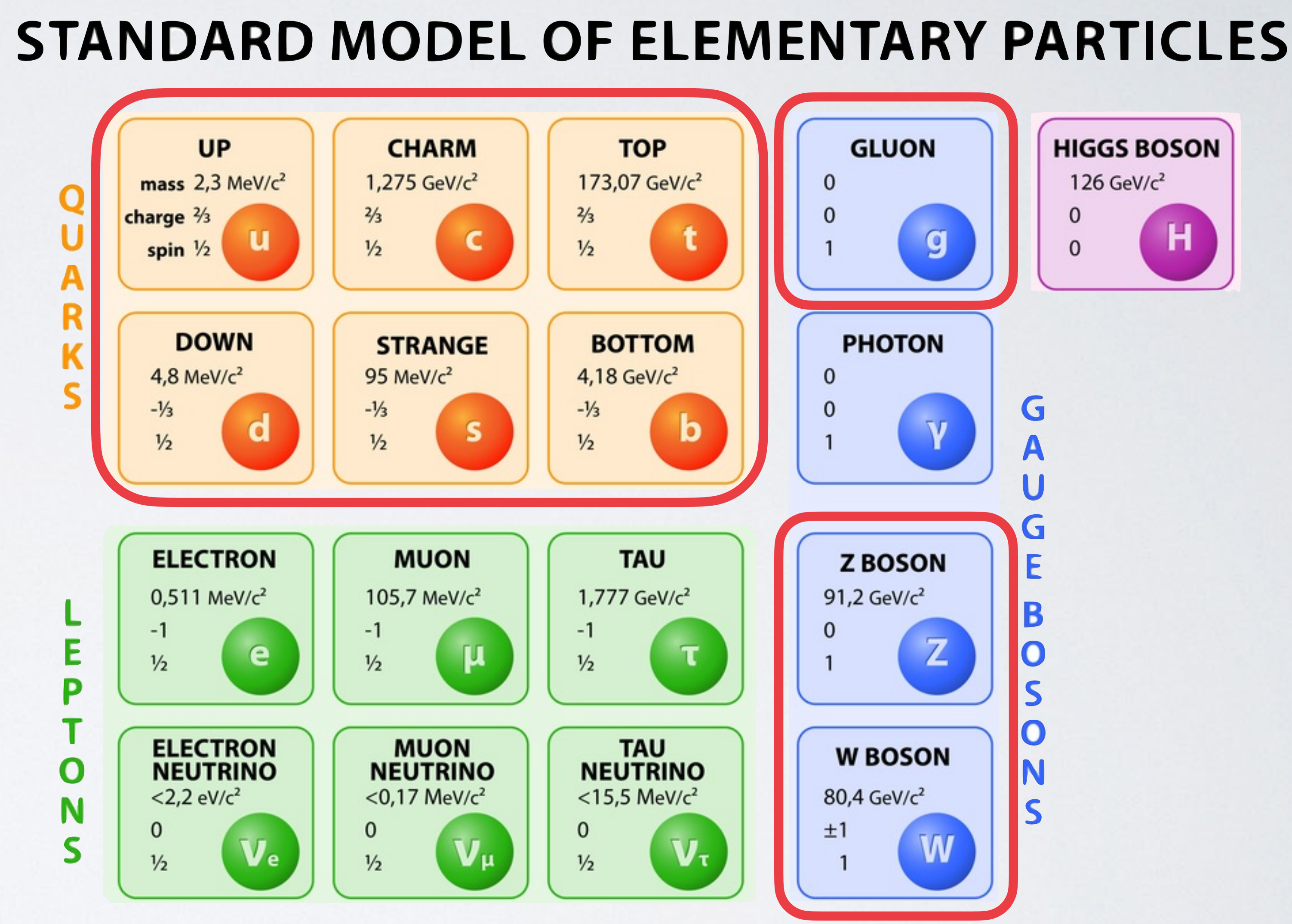
- Some GAN training techniques attempted, mostly to improve mode collapse on the MNIST Superpixels dataset (final choices in bold):
  - **Least squares**, binary cross entropy, Wasserstein, hinge and AGCD losses
  - **RMSProp**, Adam, Adadelta, SGD
  - **Two-time update rules (LR of 10^{-3} and 3×10^{-3} for G and D respectively)**
  - **Dropout (0.5 in D only)**
  - Batch Normalisation
  - Gradient Penalty
  - Spectral Normalisation
  - Noisy labels, label smoothing
  - Data augmentation
  - **Calculating quantitative metrics such as 1-Wasserstein score and Graph Fréchet Distance for model evaluation and optimisation**

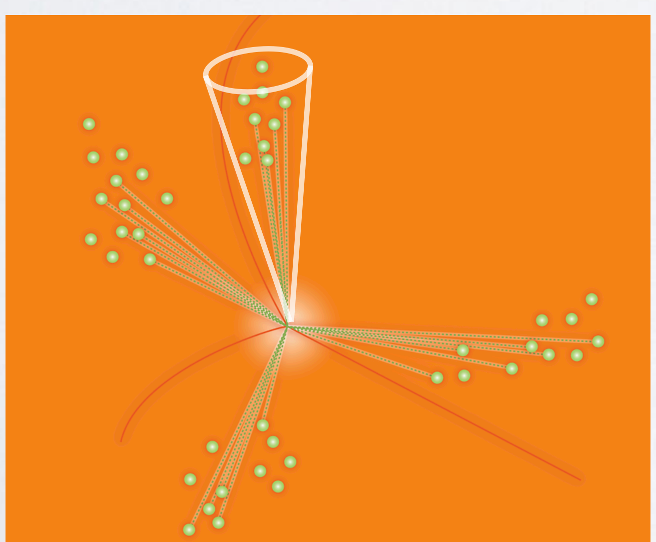
Early sample outputs of models trained on Superpixels 3s, showing clear 'mode collapse'



JETS

- Quarks, gluons, W, Z bosons are produced often at LHC but are never **directly** detected
- This is because they decay or hadronise far too quickly to reach the detector (lifetime of ~10^{-23}s)
- Instead after decaying/hadronising (multiple times sometimes) they produce a set of stable particles with distinct features and geometries, called a jet
- From this we an infer the originating particle

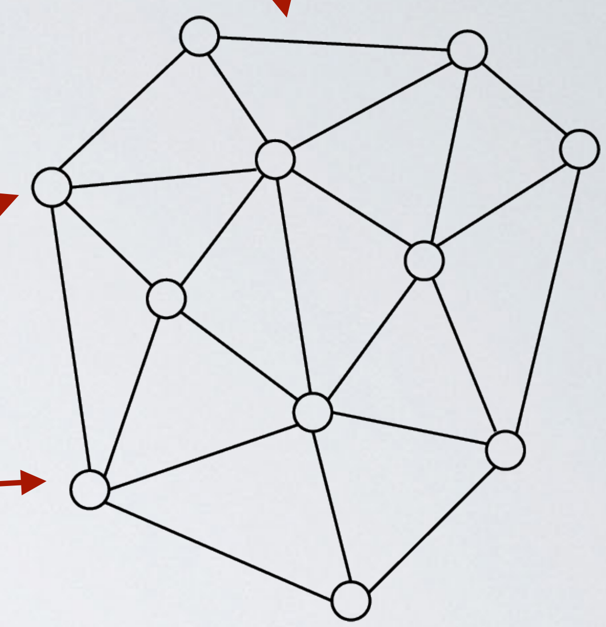


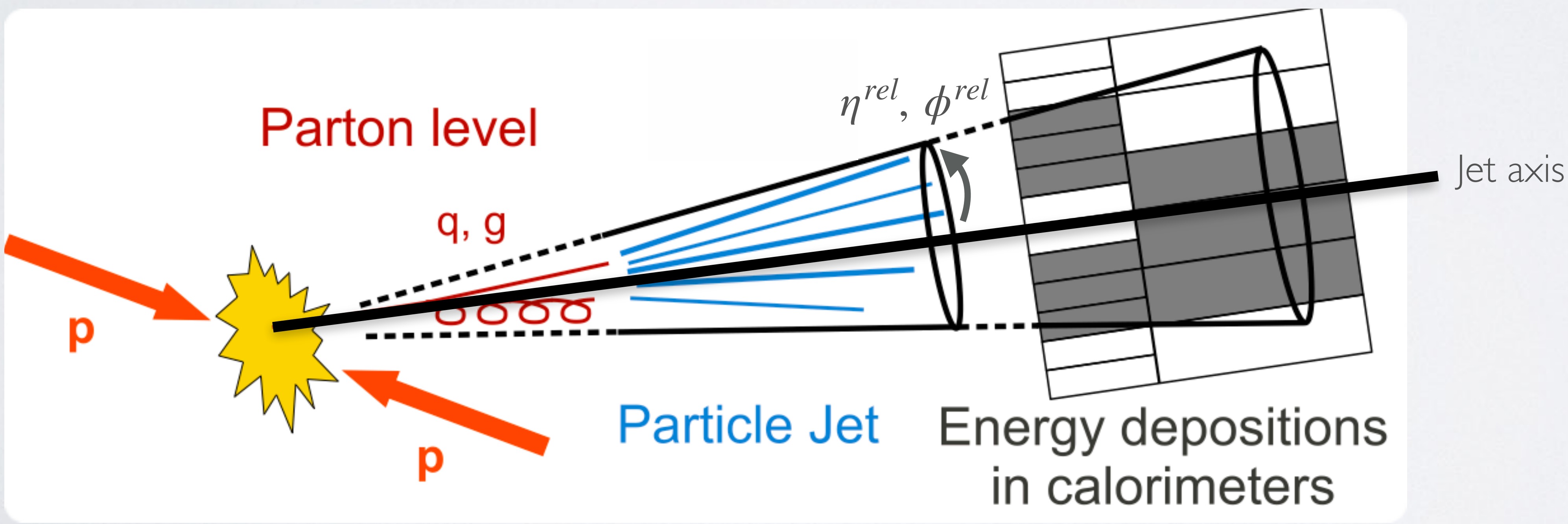


JETS

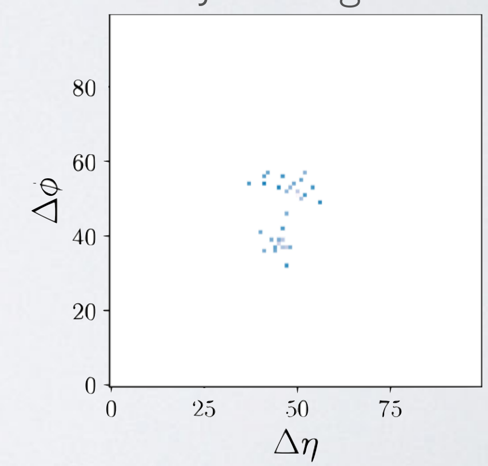
Edge features are angular distances $R = \sqrt{\eta^2 + \phi^2}$ between nodes

- Simulated high transverse momentum (p_T) jets
- In each jet we consider the N highest p_T particle constituents (zero padding when needed), and 3 particle features (η^{rel}, ϕ^{rel}, p_T^{rel})



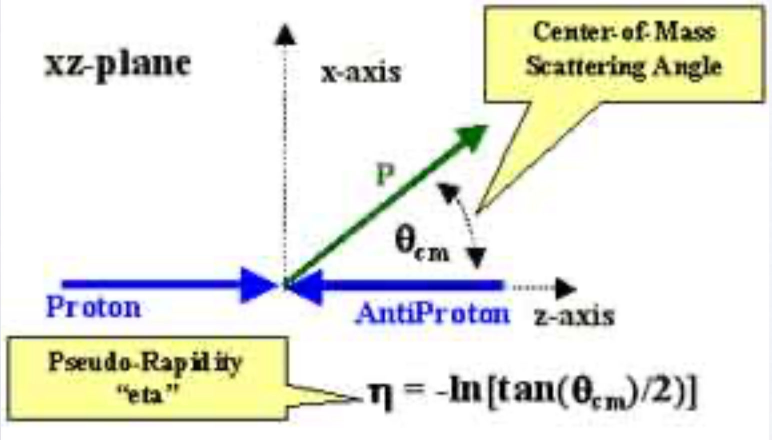


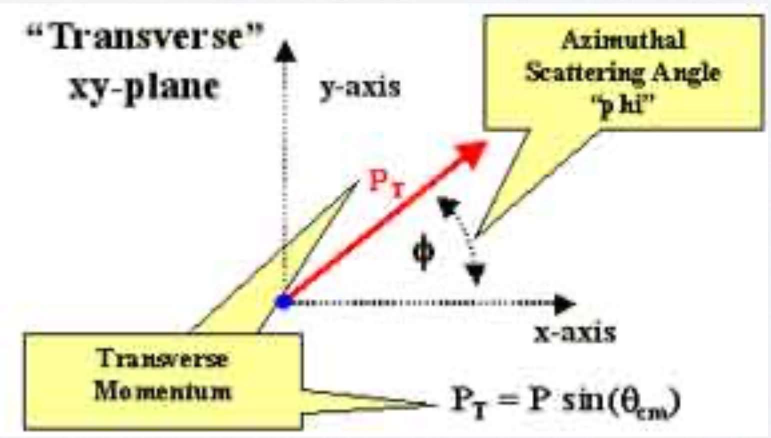
Jet 'image'

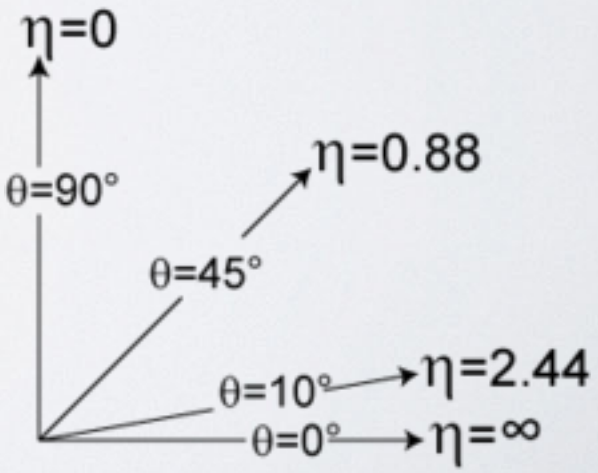


LHC GEOMETRY

- Variables are chosen carefully to be invariant to 'boosts' along the 'beam-line' (z-axis), so we can always boost, or switch, to the centre of mass or some other convenient frame of the collision
- Instead of momentum p, *transverse* momentum p_T
- Instead of $\theta, \eta = -\ln\left[\tan(\theta/2)\right]$ $\Rightarrow$ that $\Delta\eta$ is invariant to boosts so we can freely translate in η space (also defined so that the angle perpendicular to beam axis $\theta = \pi/2 \rightarrow \eta = 0$)







EVALUATION: JETS

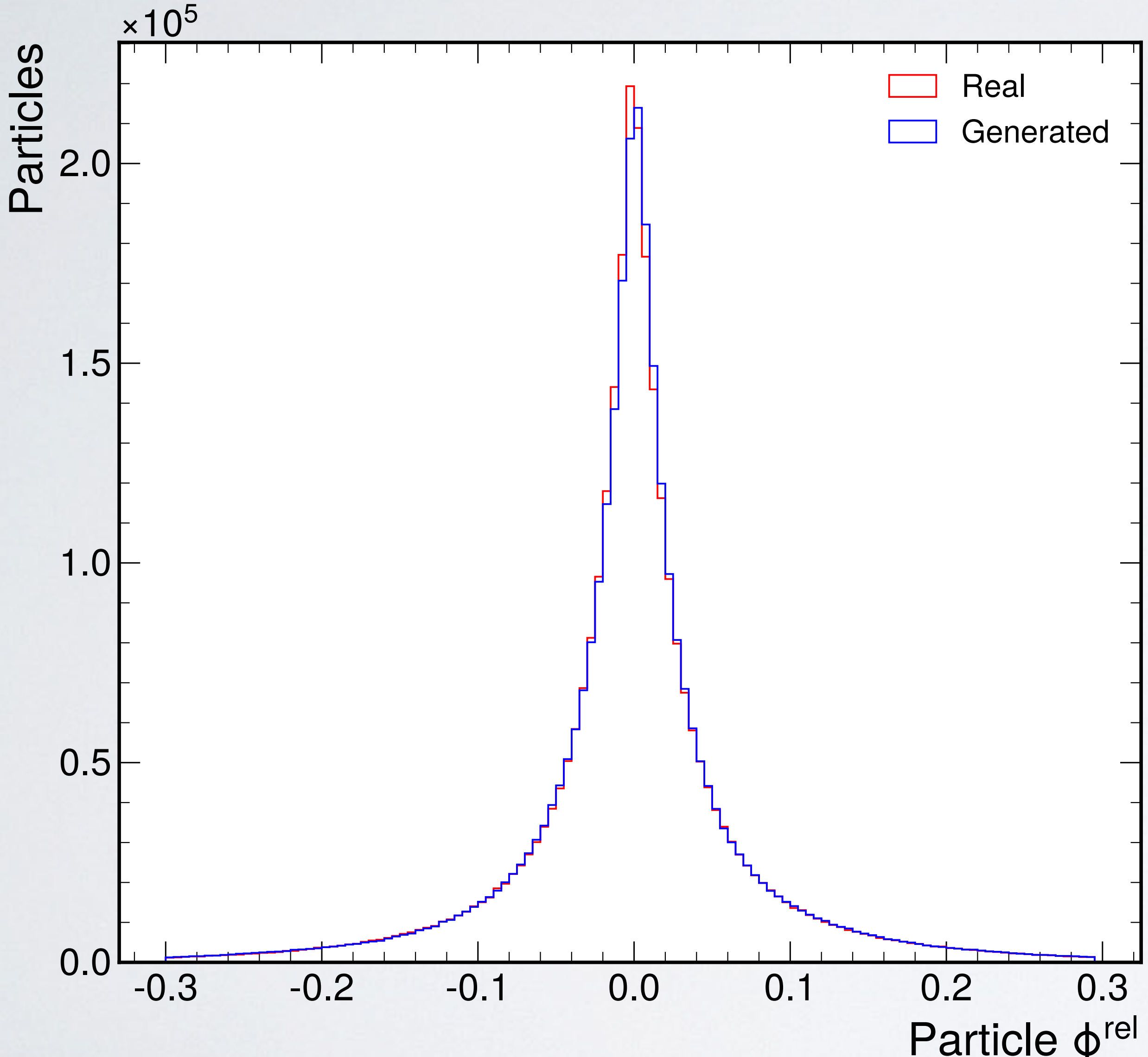
- We look at particle η^{rel}, ϕ^{rel}, p_T^{rel}, and jet mass and Energy Flow Polynomials (Komiske et al.) distributions
  - EFPs calculate n-particle correlations per jet
  - Span the set of useful* jet observables
- We use 1-Wasserstein (W_1) metric (minimum work needed to transform one dist. to another)
  - Baseline: average W_1 between different sets of samples of real jets
  - Score: average W_1 between sets of real and fake samples

*i.e. IRC safe

RESULTS: JETS (PARTICLE LEVEL)

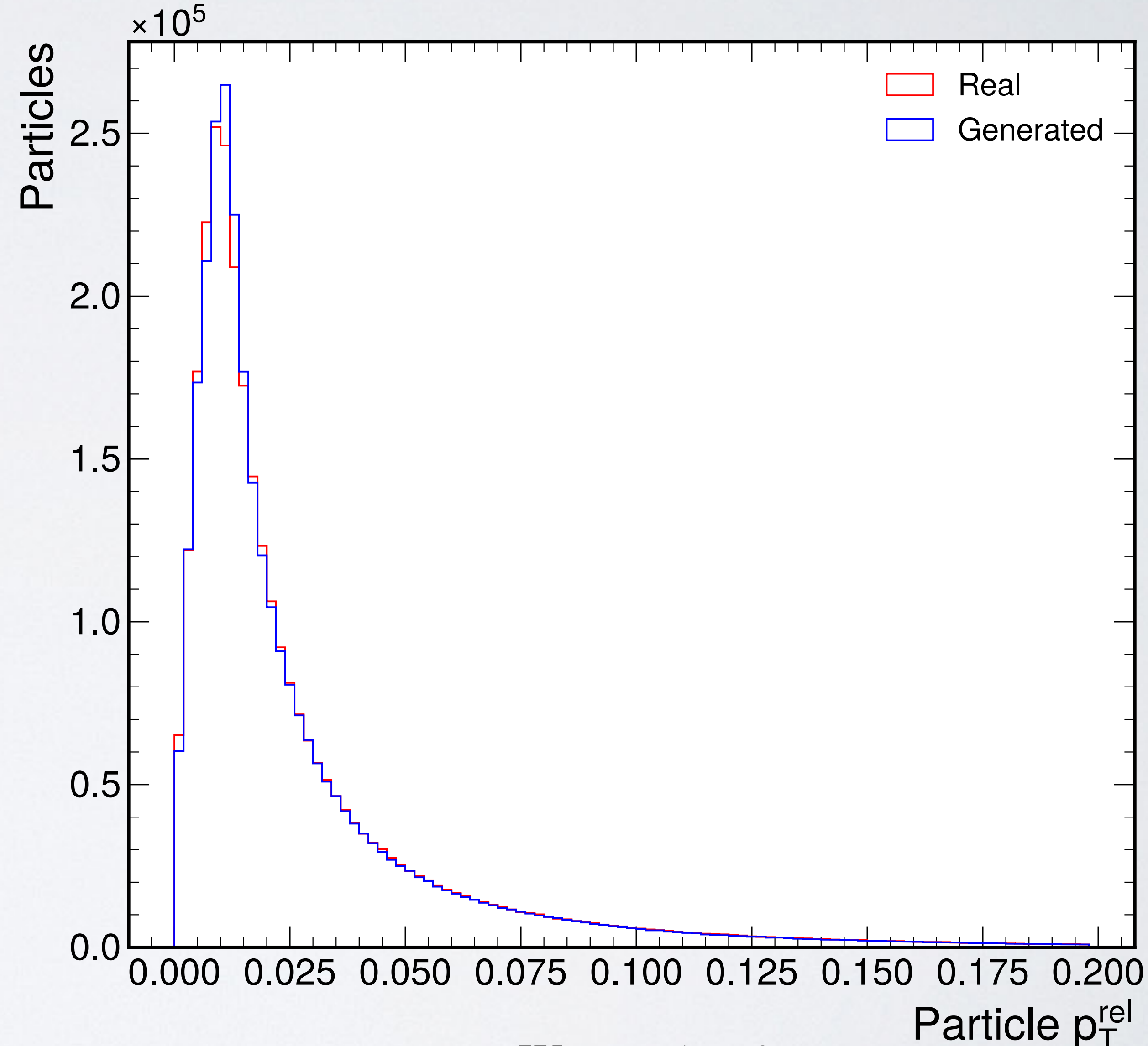
Jets Dataset (Up to 30 particles per **gluon** jets of $p_T \sim 1\text{TeV}$)

- Real distributions are reproduced with high fidelity:



Real vs Real $W_1 = 6 \pm 2^*$

Real vs Fake $W_1 = 11 \pm 4$



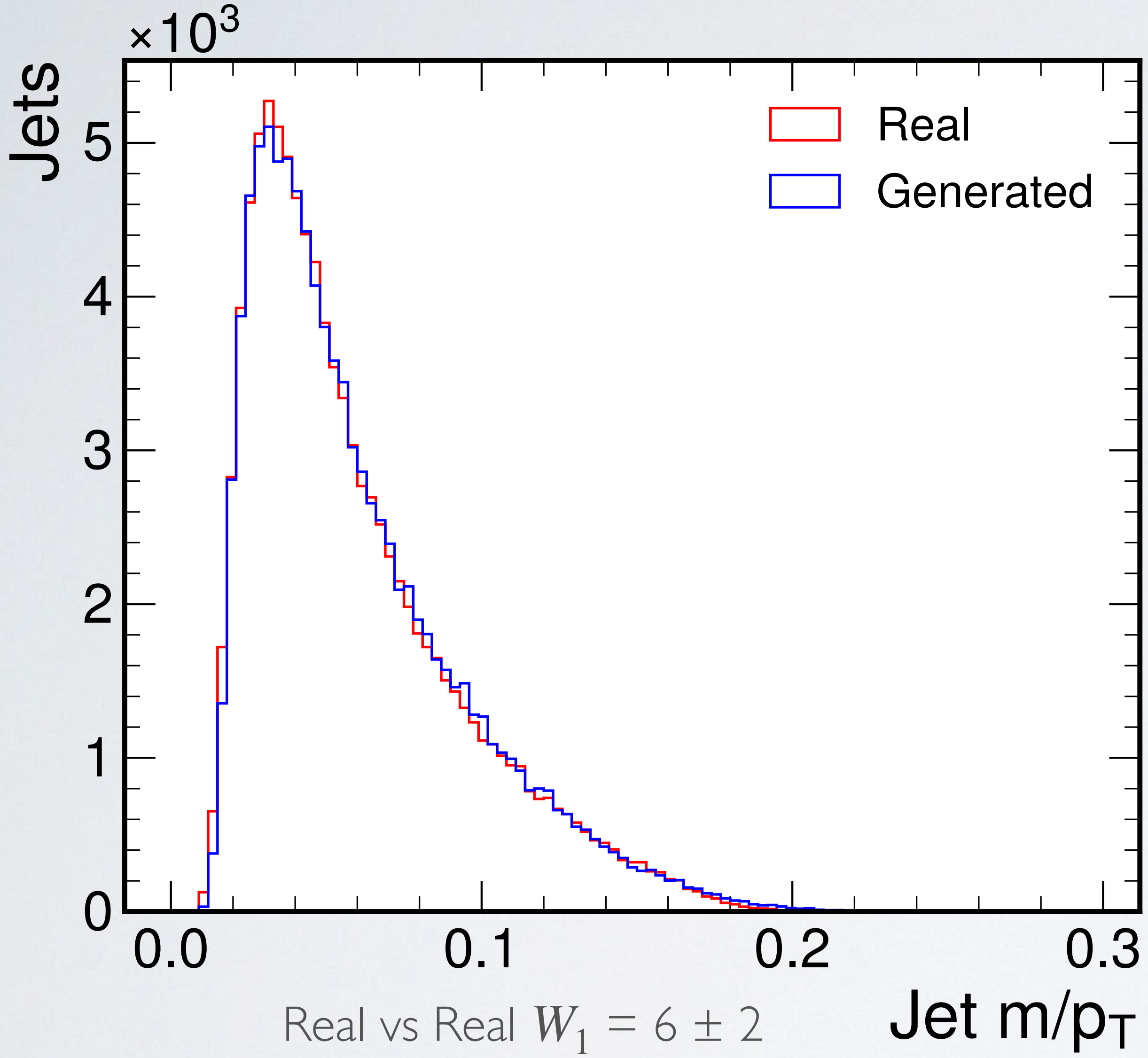
Real vs Real $W_1 = 1.4 \pm 0.5$

Real vs Fake $W_1 = 2 \pm 1$

*All scores scaled by 10^3

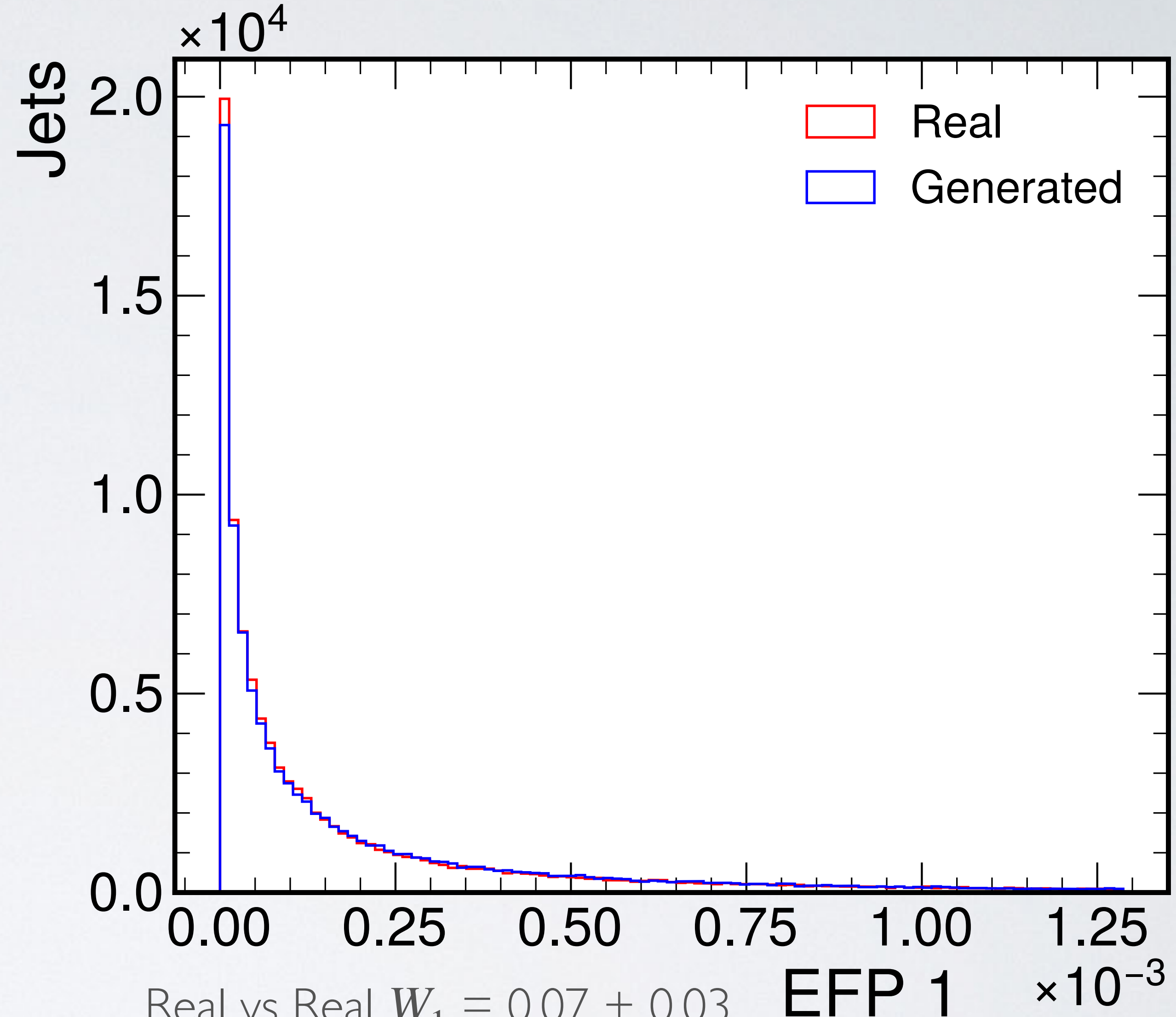
RESULTS: JETS (JET LEVEL)

- Real distributions are reproduced with high fidelity (4-particle correlation EFP):



Real vs Real $W_1 = 6 \pm 2$

Real vs Fake $W_1 = 6 \pm 2$



Real vs Real $W_1 = 0.07 \pm 0.03$

Real vs Fake $W_1 = 0.08 \pm 0.03$

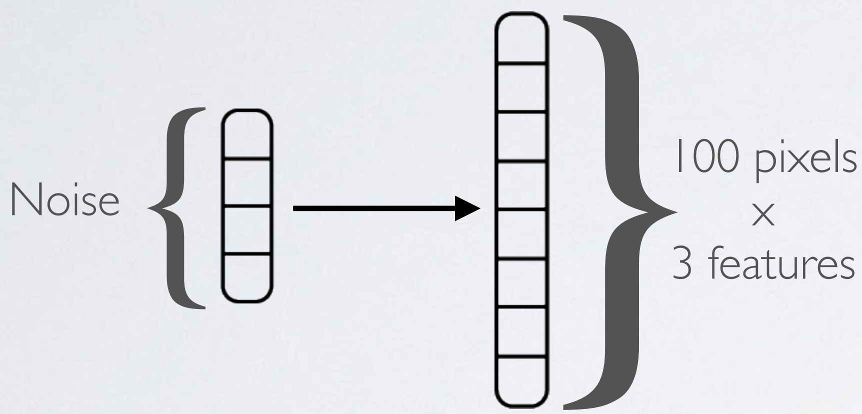
- Remarkable that without specific guidance complex physics is learnt
- Learning tails well ⇒ little evidence of mode collapse

GAN ARCHITECTURES (NON-GRAPH)

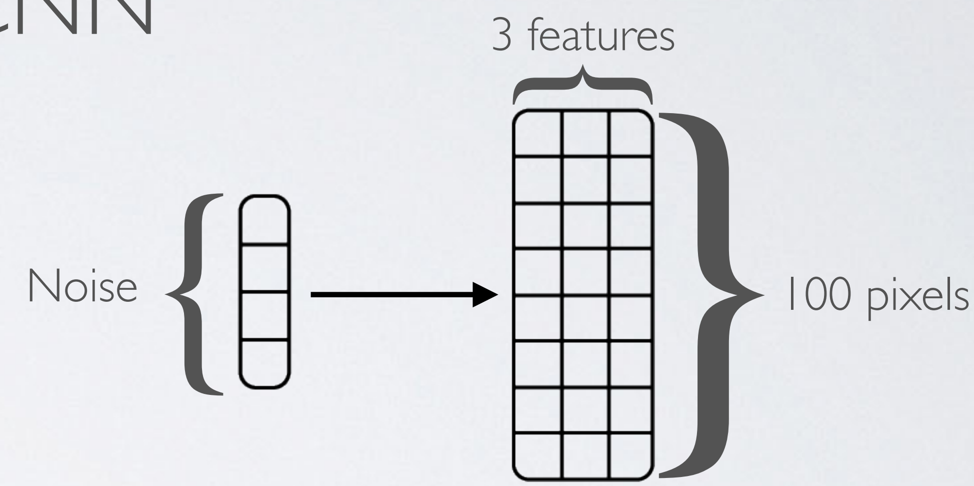
- First, baseline attempt at using our sparse data structure with standard NN and CNN based GANs:

Generator

Neural Network

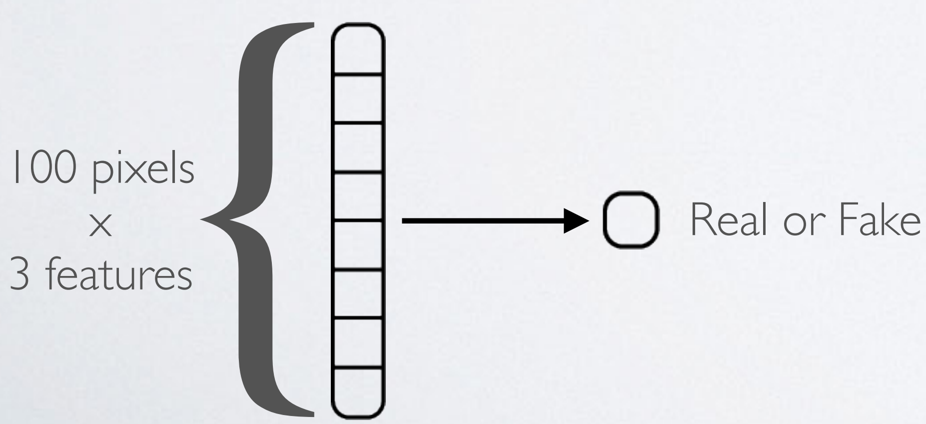


CNN

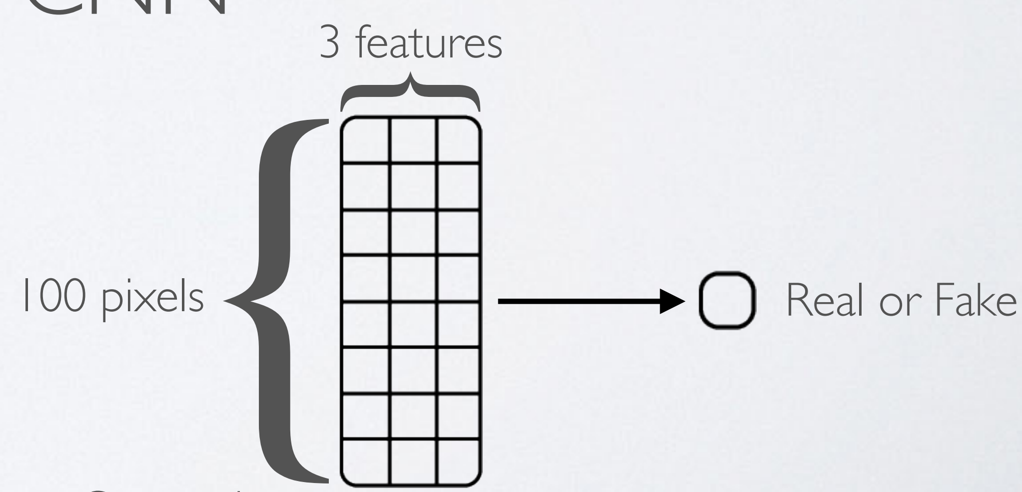


Discriminator

Neural Network

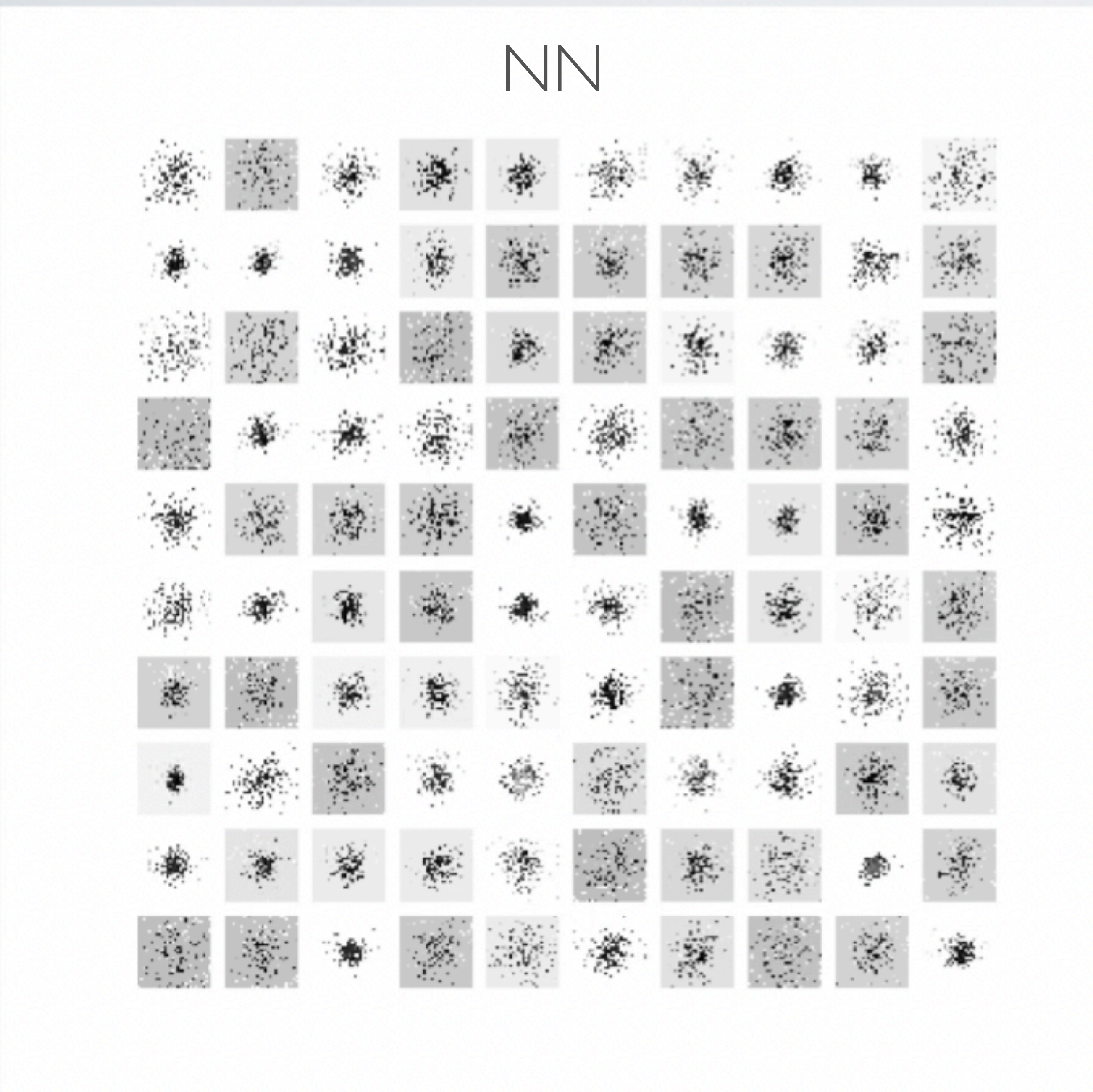


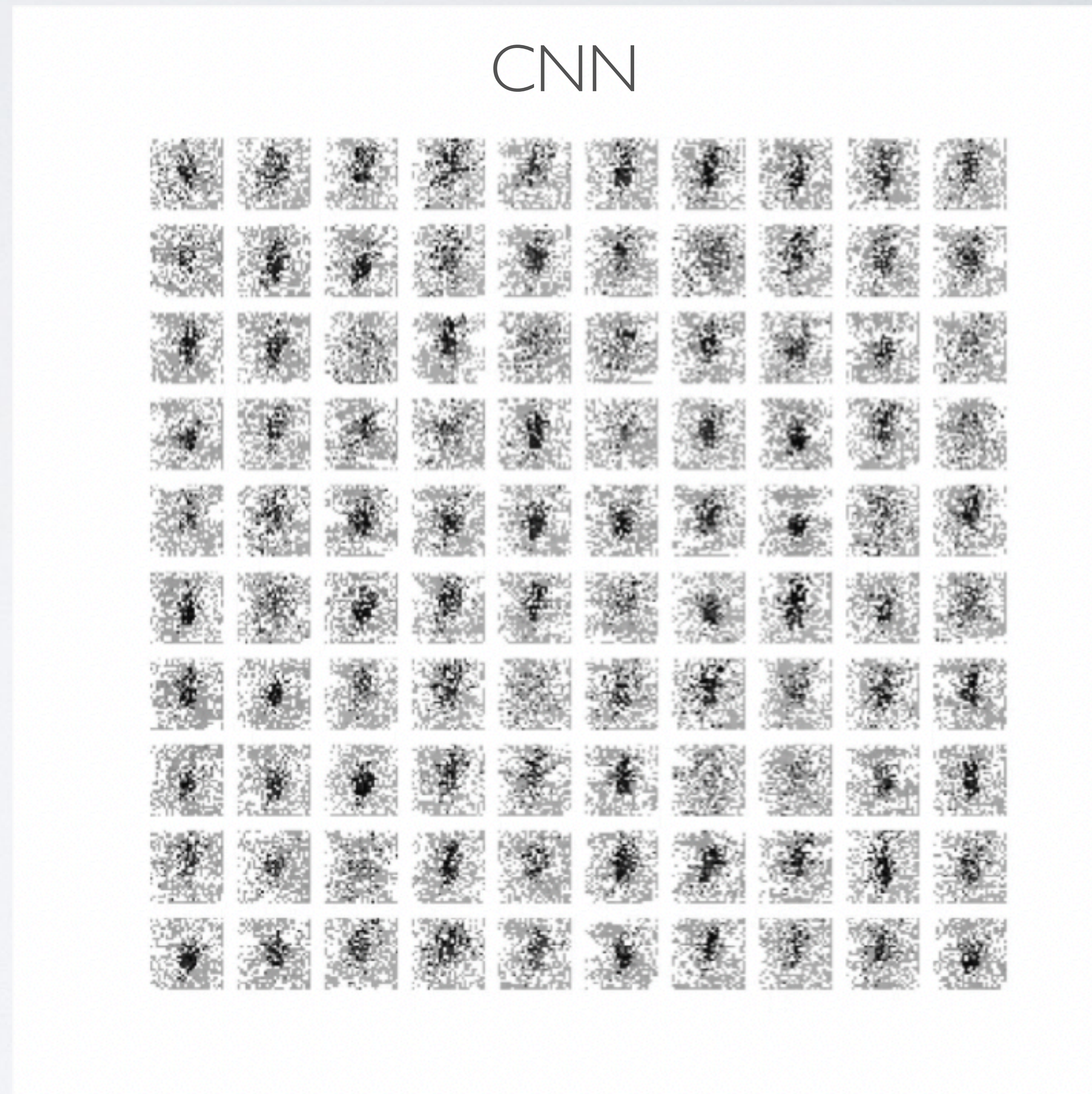
CNN



GAN ARCHITECTURES (NON-GRAPH)

Outputs (on Sparse MNIST) were just noise:

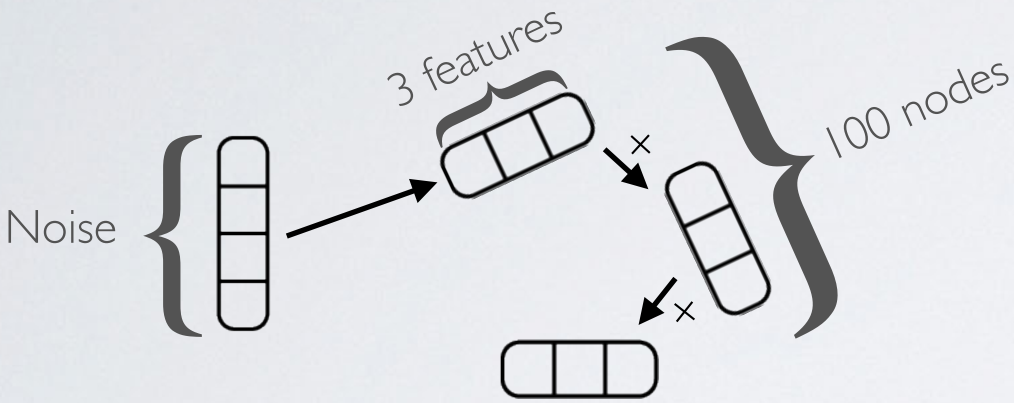




GAN ARCHITECTURES

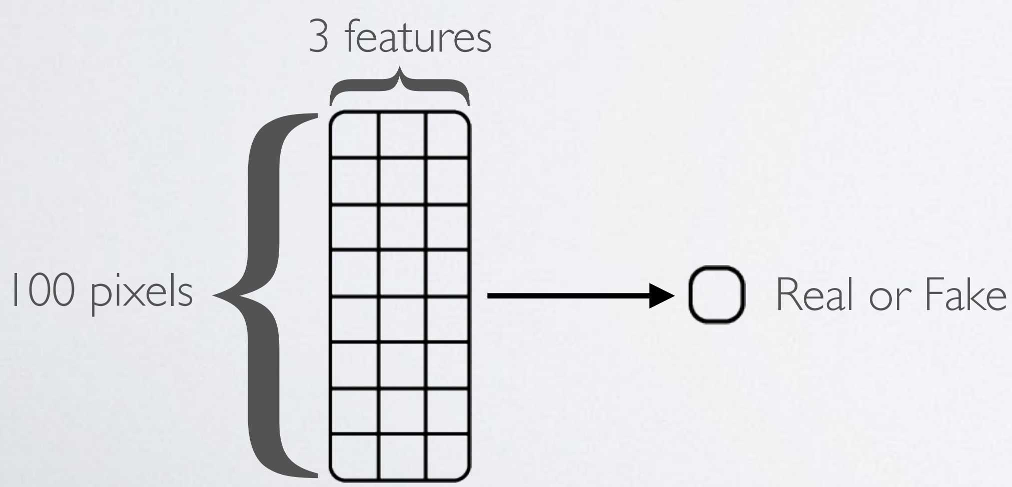
Generator

Node generation via RNN



Discriminator

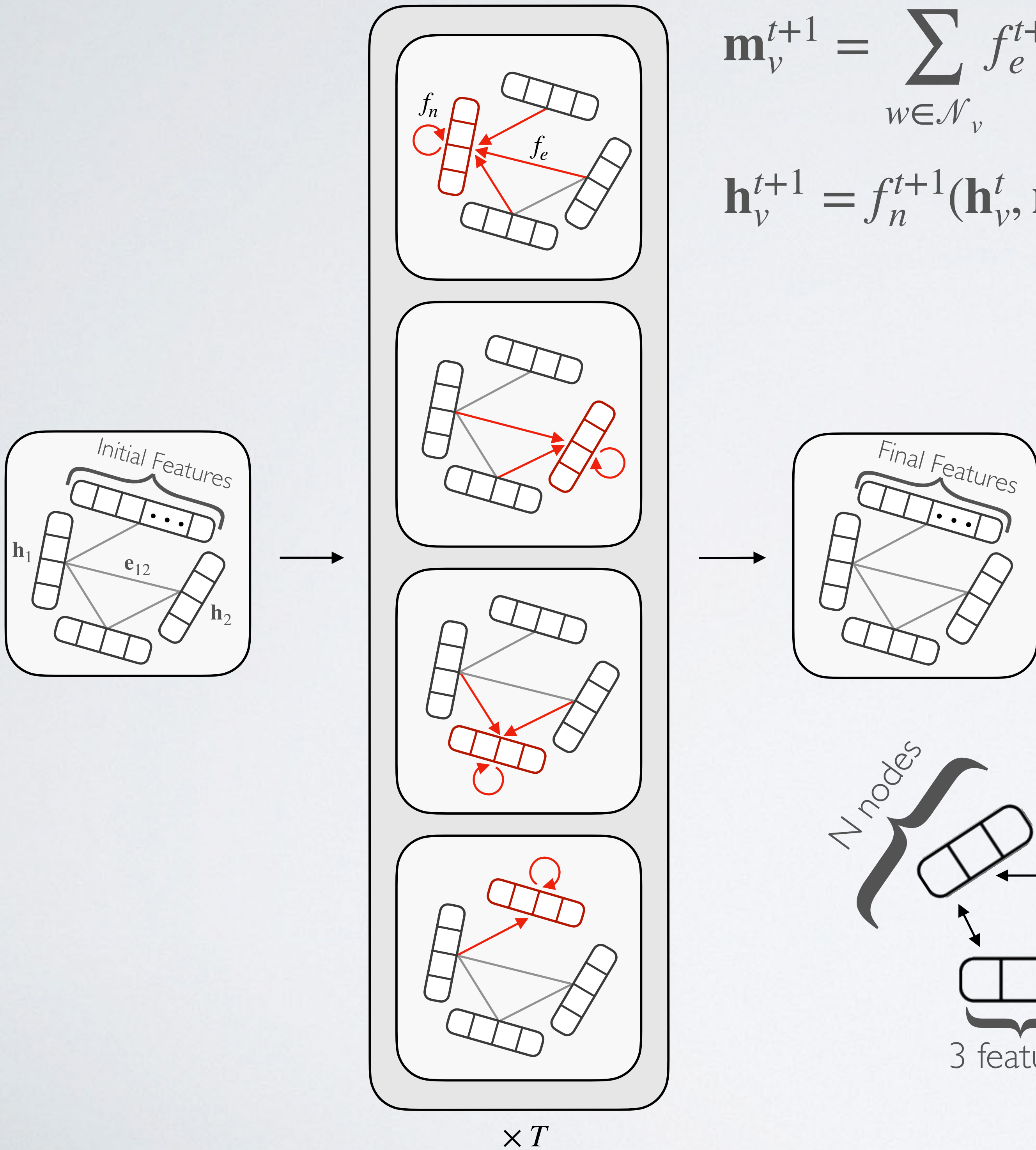
CNN



- Next, recurrent neural network (RNN) based generator which can produce graphs with arbitrary numbers of nodes
- But while outputs (on Sparse MNIST, below) showed learning of some structure, this architecture was unable to reproduce real samples

GAN ARCHITECTURE

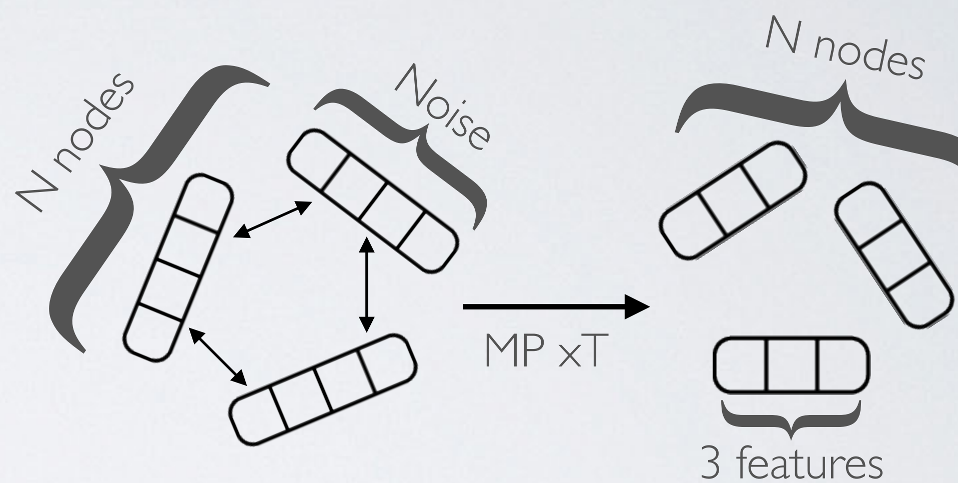
End goal is a generator which takes graph/jet-level variables into account, so we try a modified 'conditional' GAN/generalised MPNN:



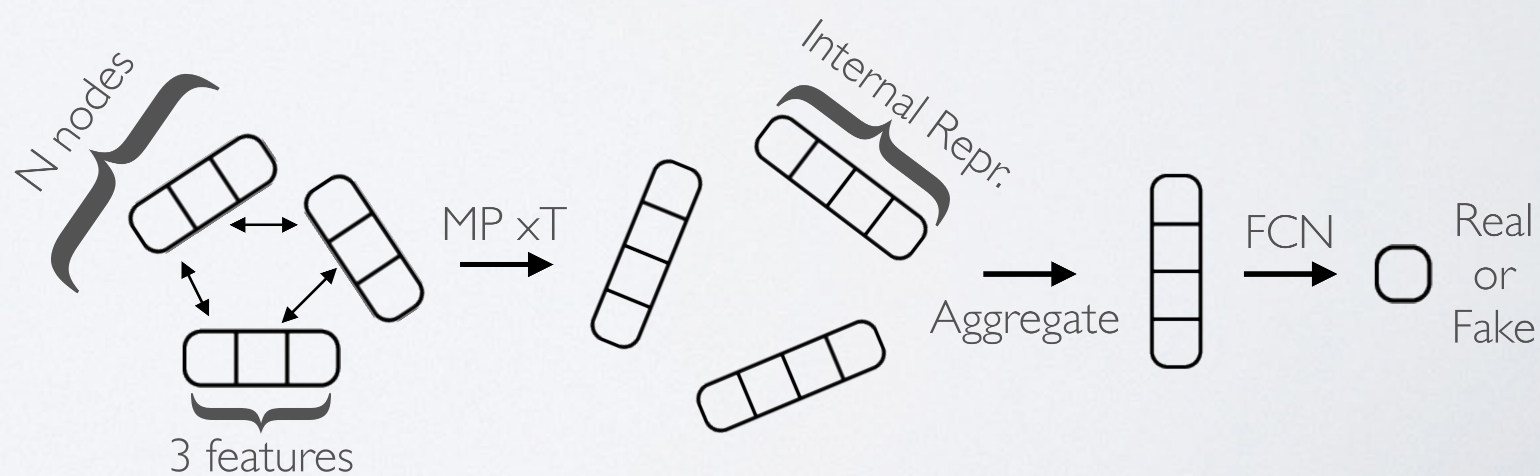
$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1})$$

Generator



Discriminator

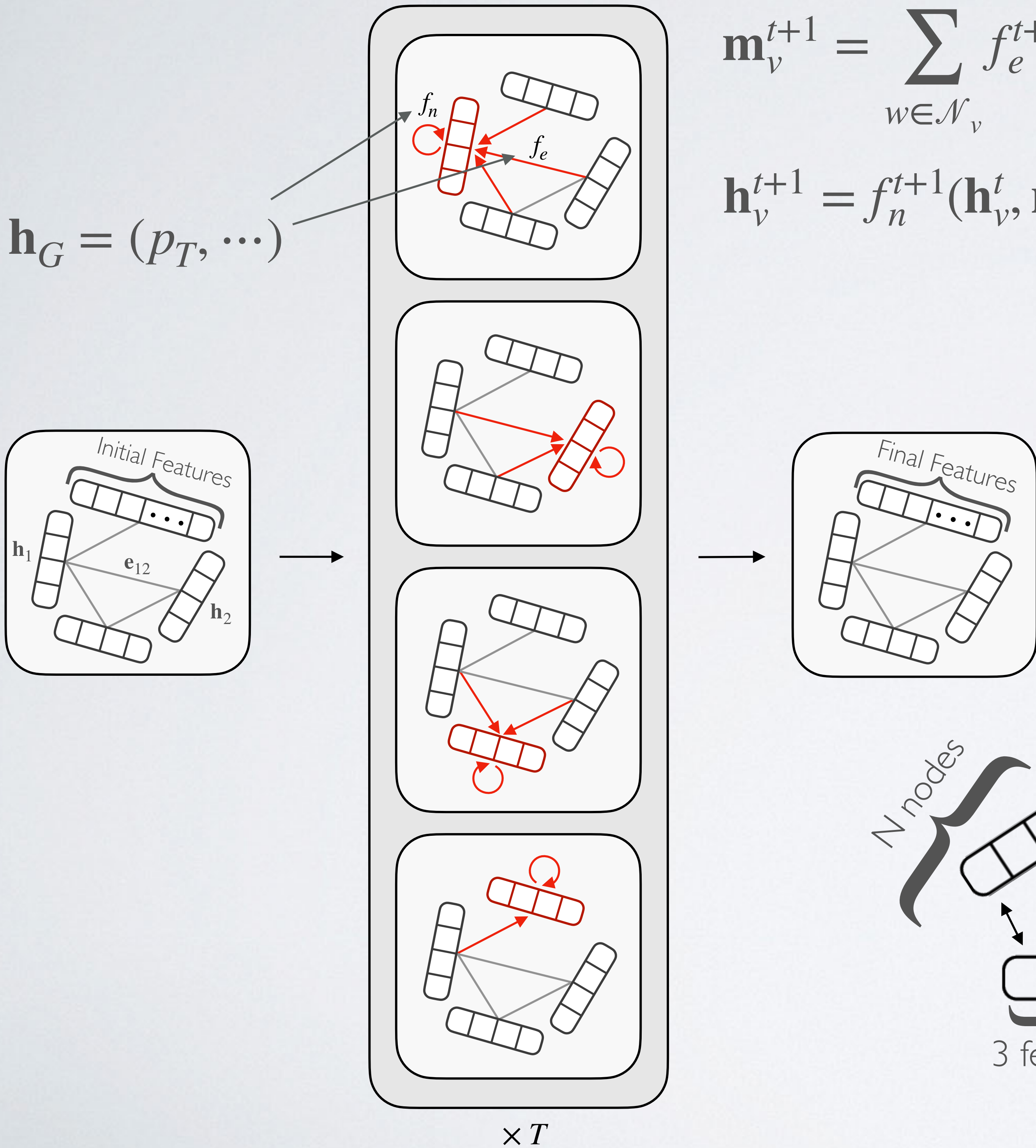


GAN ARCHITECTURE

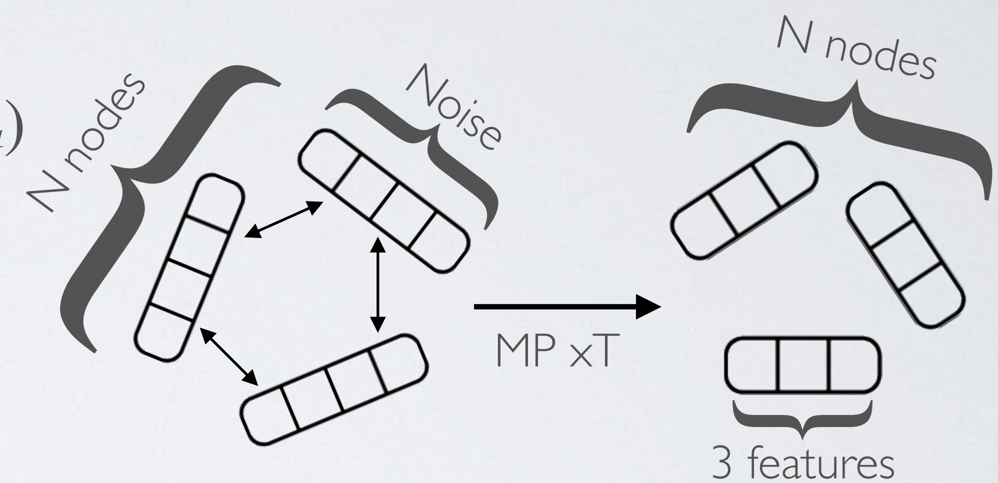
End goal is a generator which takes graph/jet-level variables into account, so we try a modified 'conditional' GAN/generalised MPNN:

$$\mathbf{m}_v^{t+1} = \sum_{w\in\mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

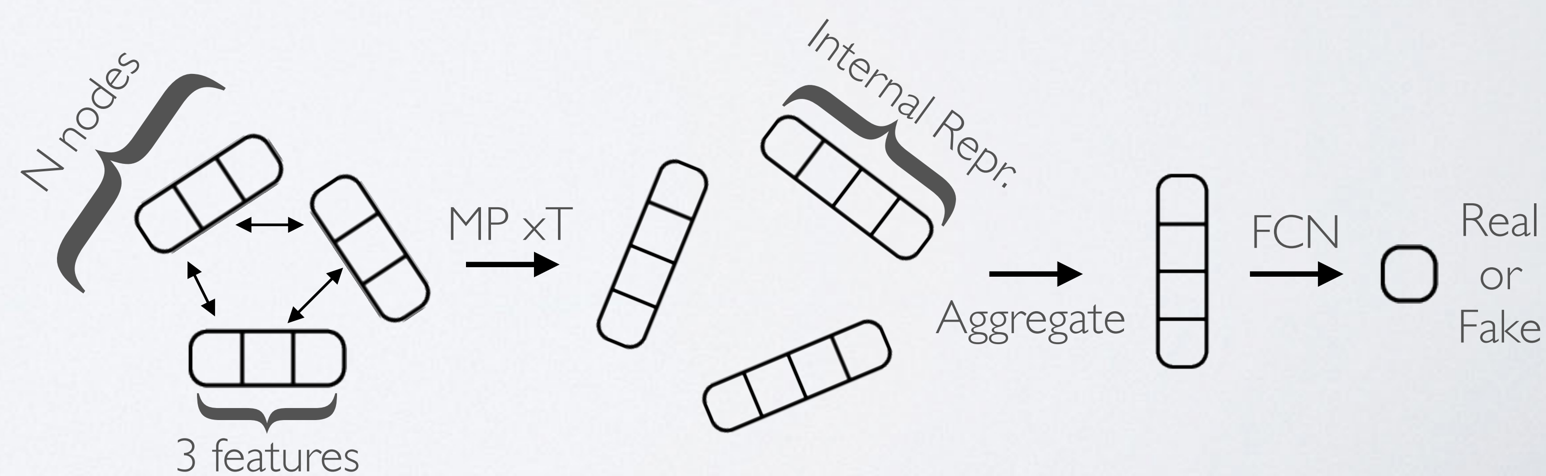
$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$



Generator



Discriminator

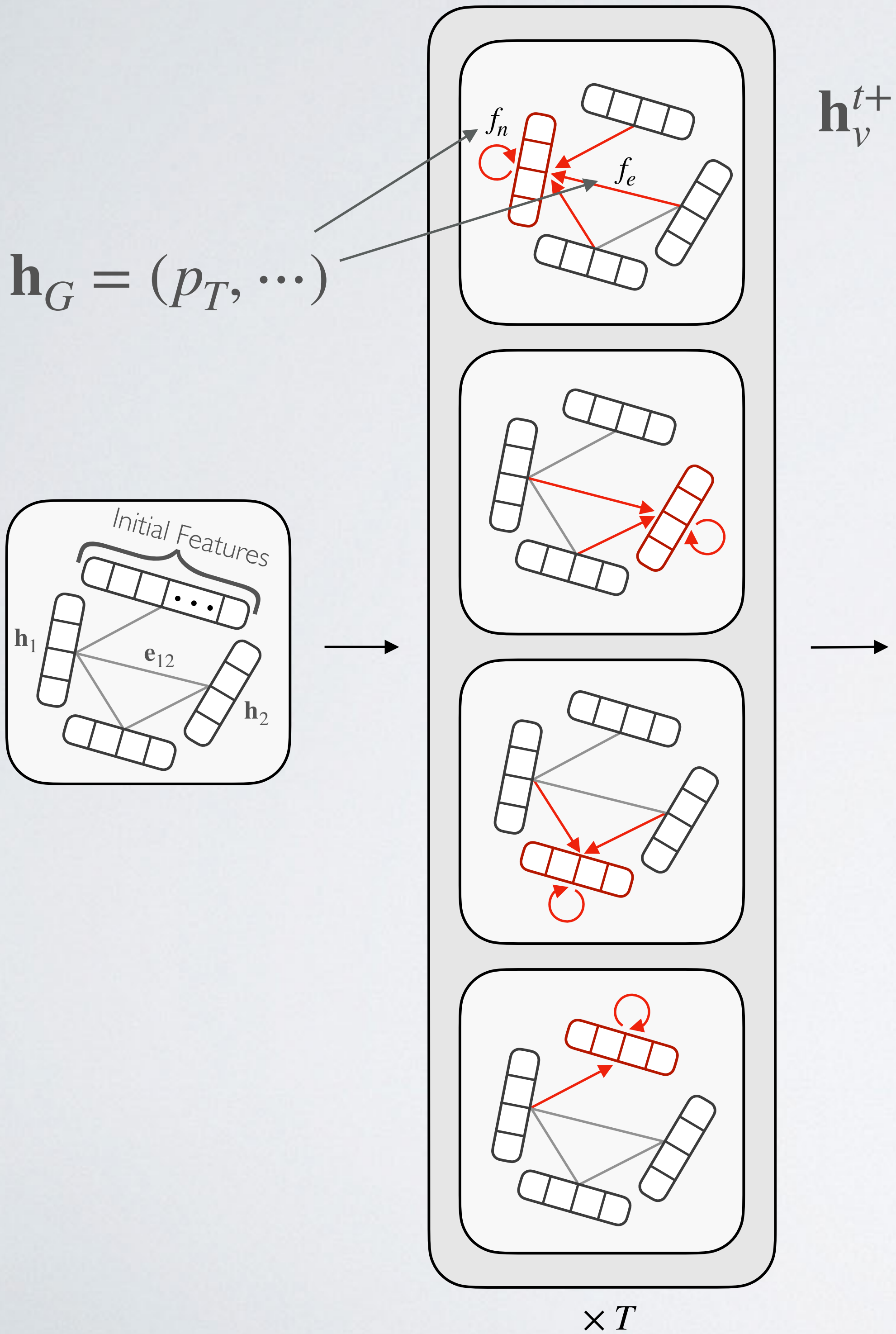


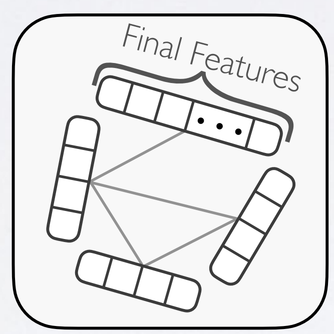
GAN ARCHITECTURE

To allow for variable sized graphs we 'mask' extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$



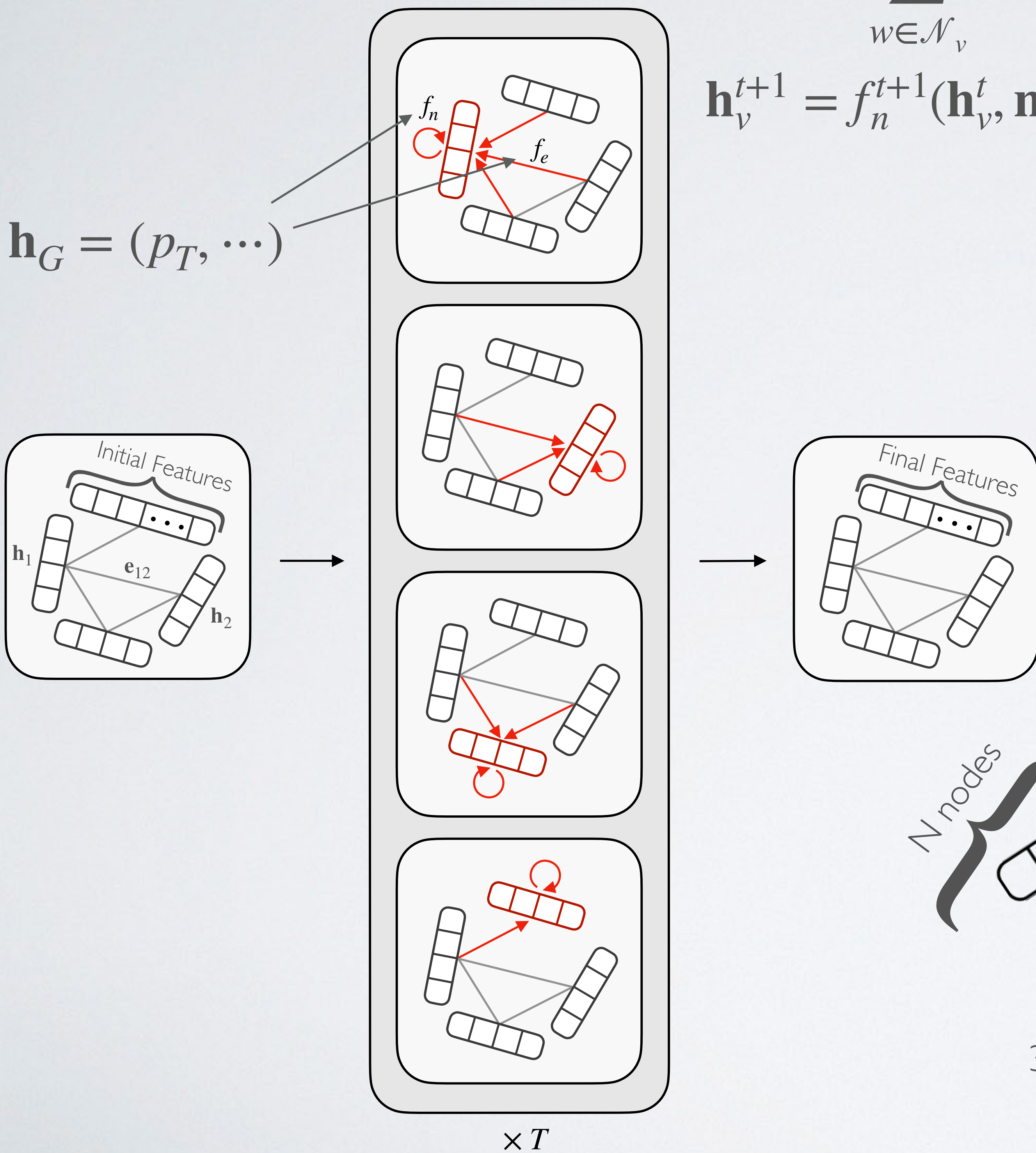


GAN ARCHITECTURE

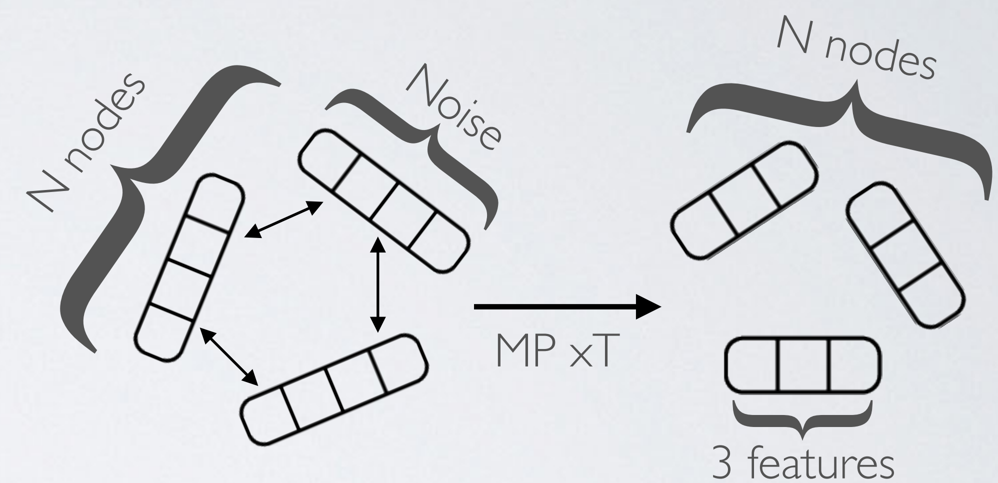
To allow for variable sized graphs we 'mask' extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

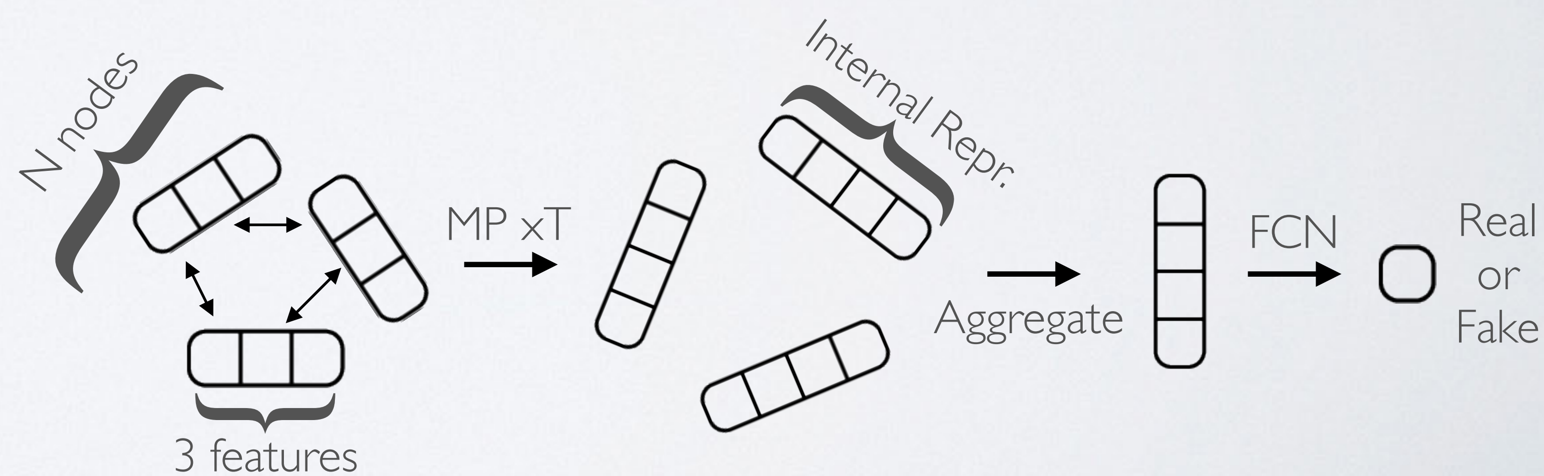
$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$



Generator



Discriminator

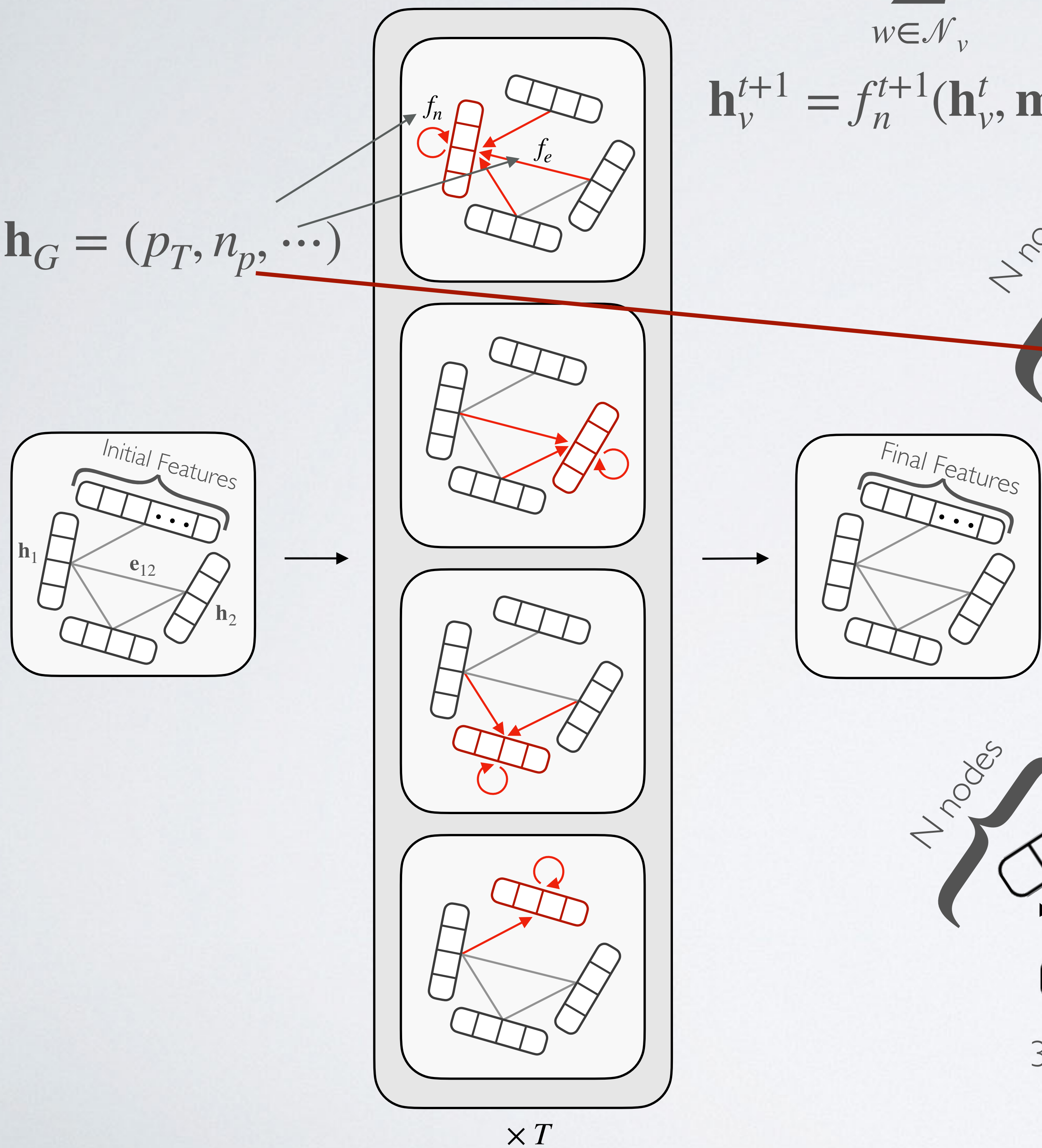


GAN ARCHITECTURE

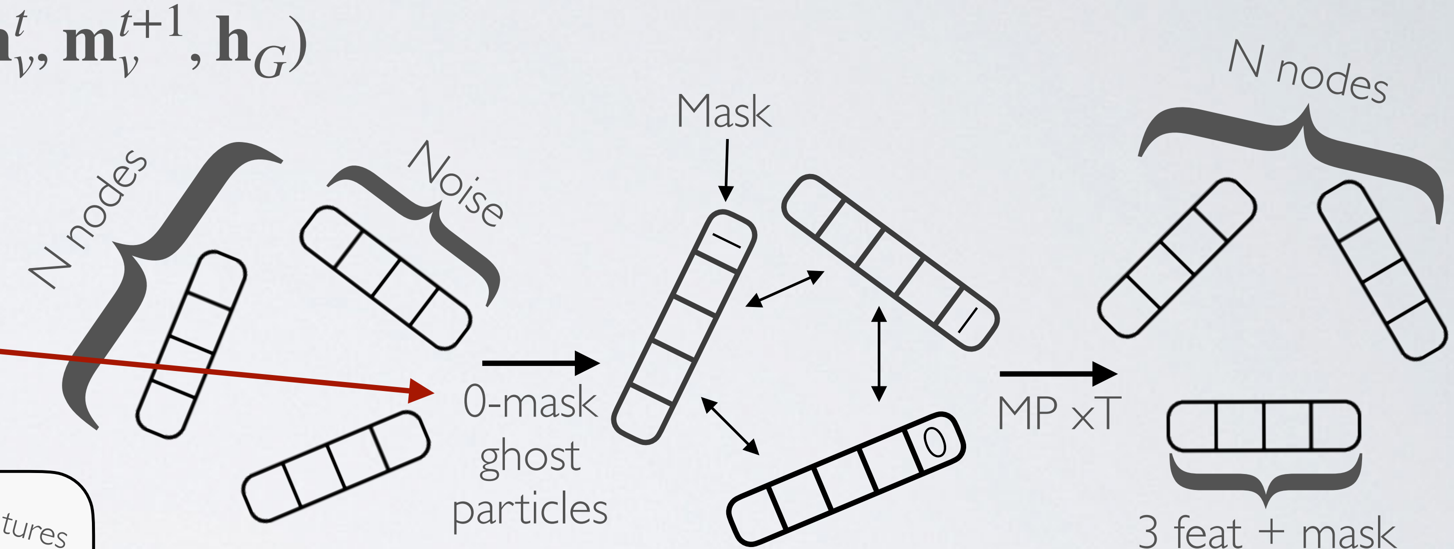
To allow for variable sized graphs we 'mask' extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

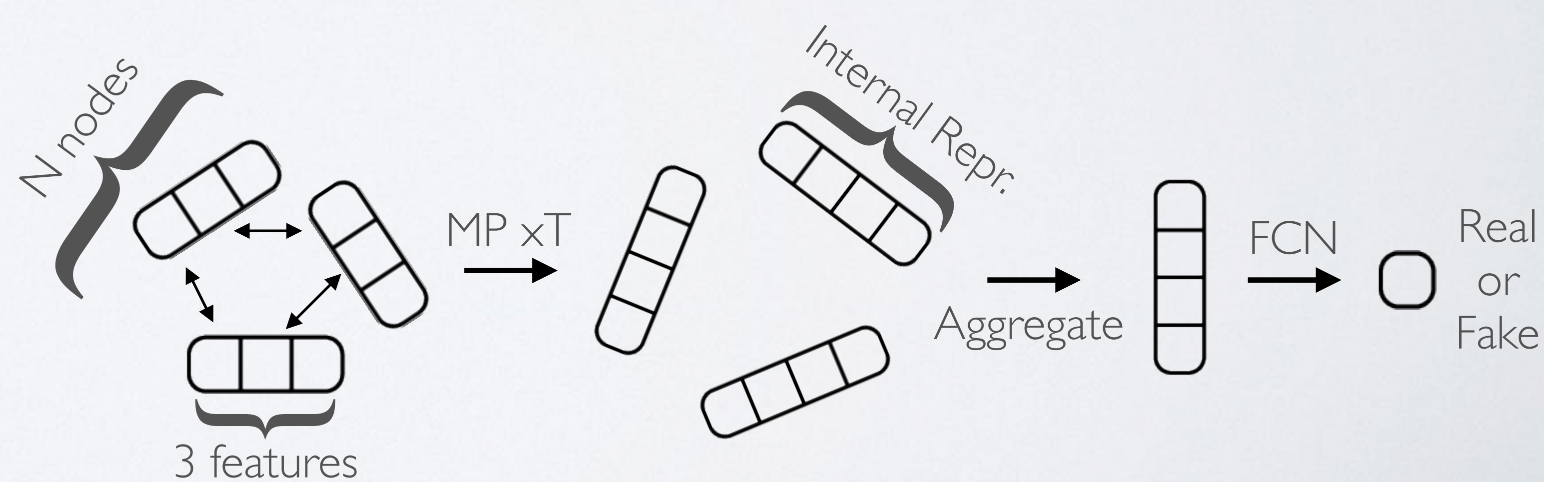
$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$



Generator



Discriminator

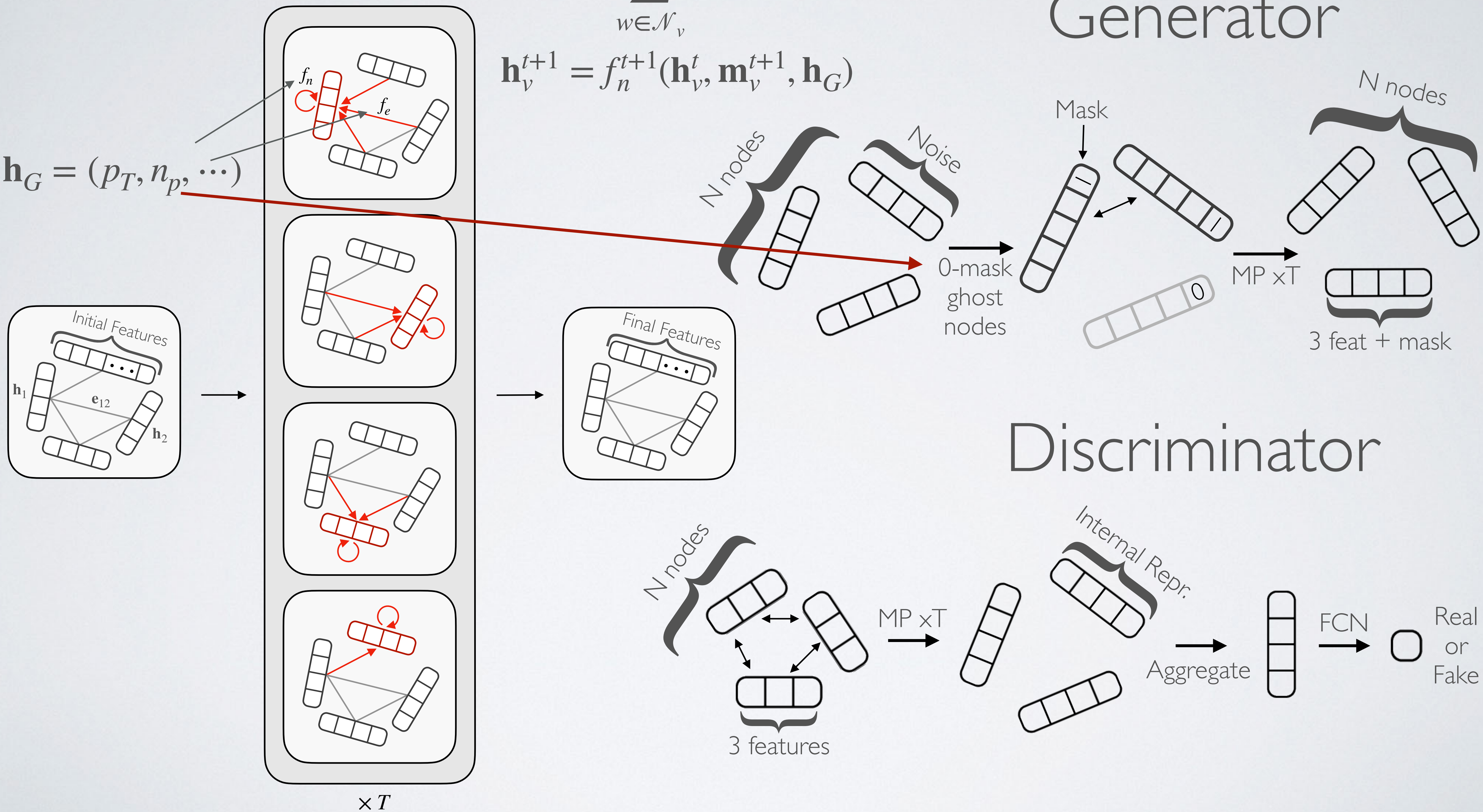


GAN ARCHITECTURE

To allow for variable sized graphs we 'mask' extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$

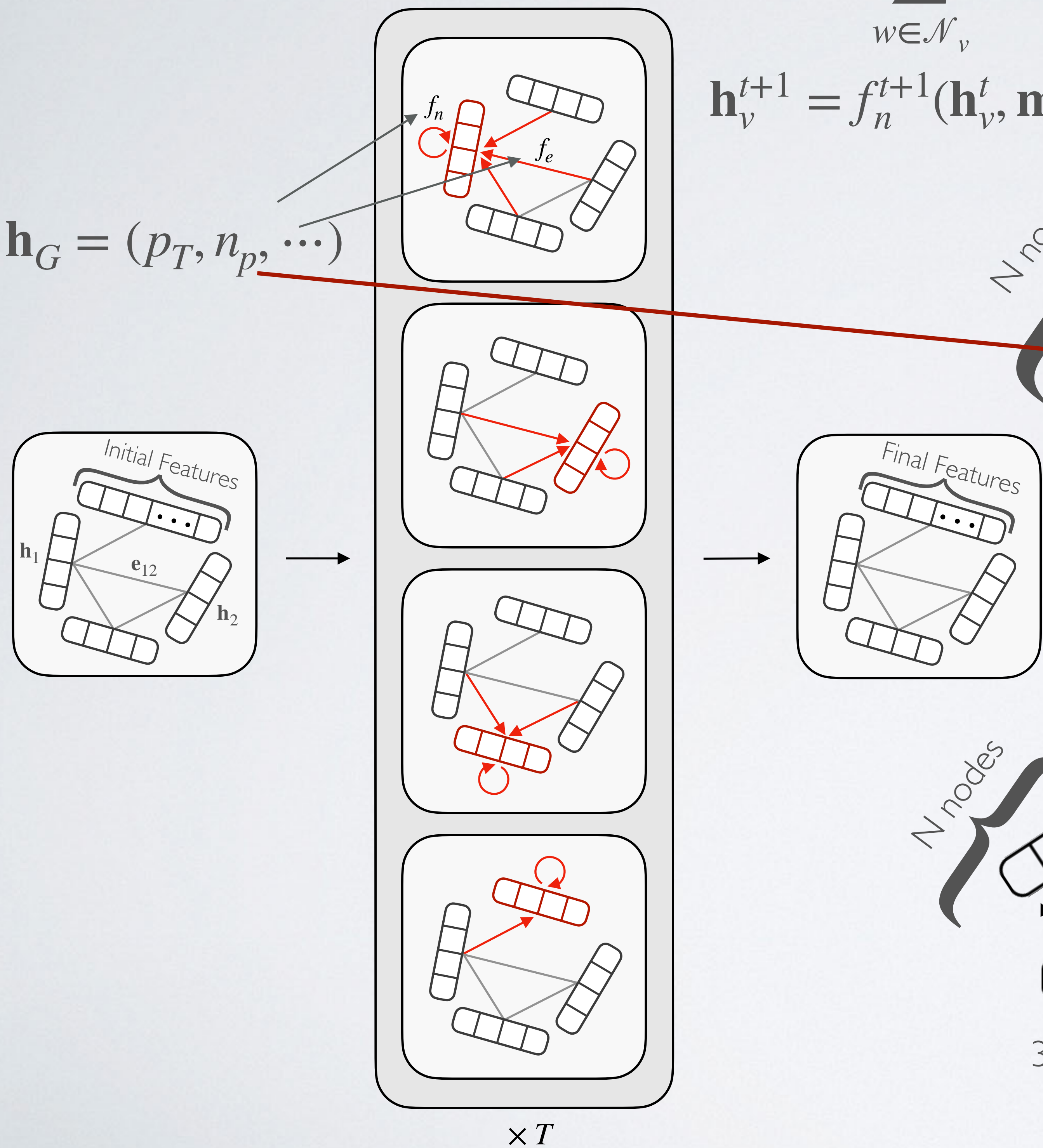


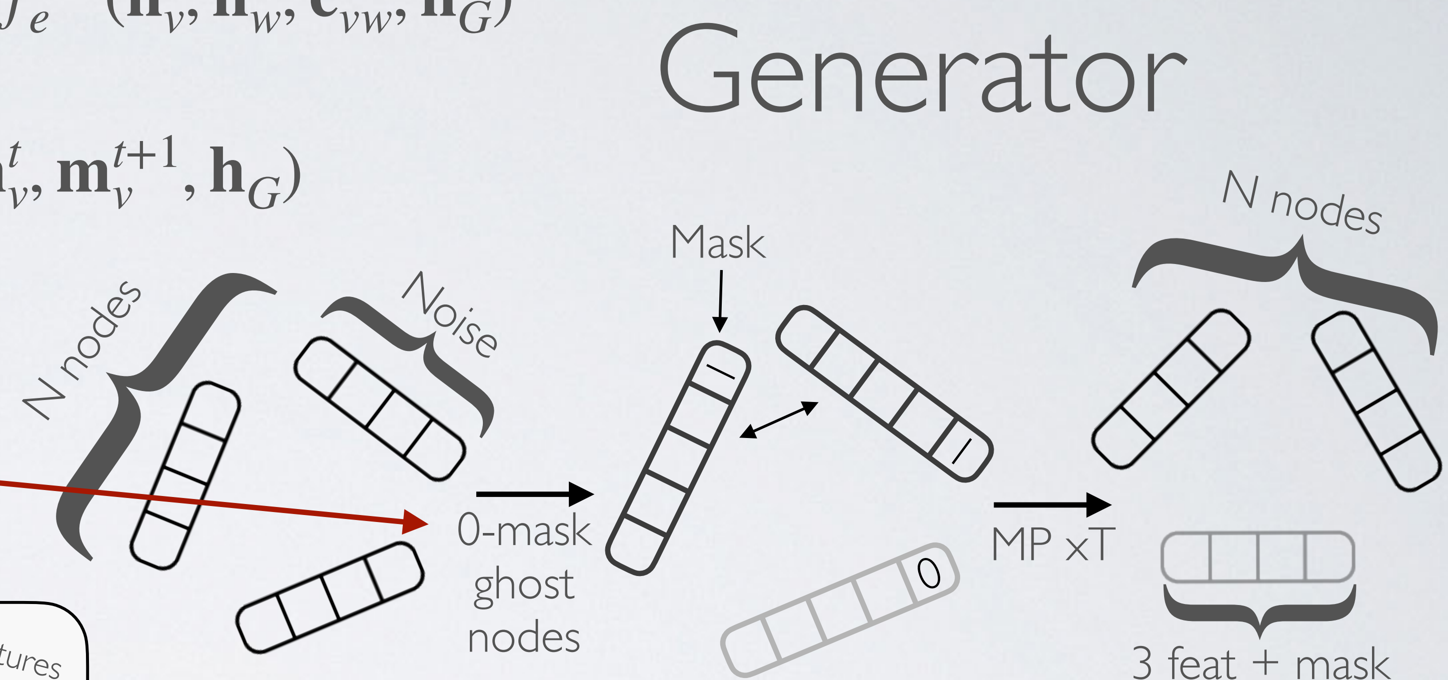
GAN ARCHITECTURE

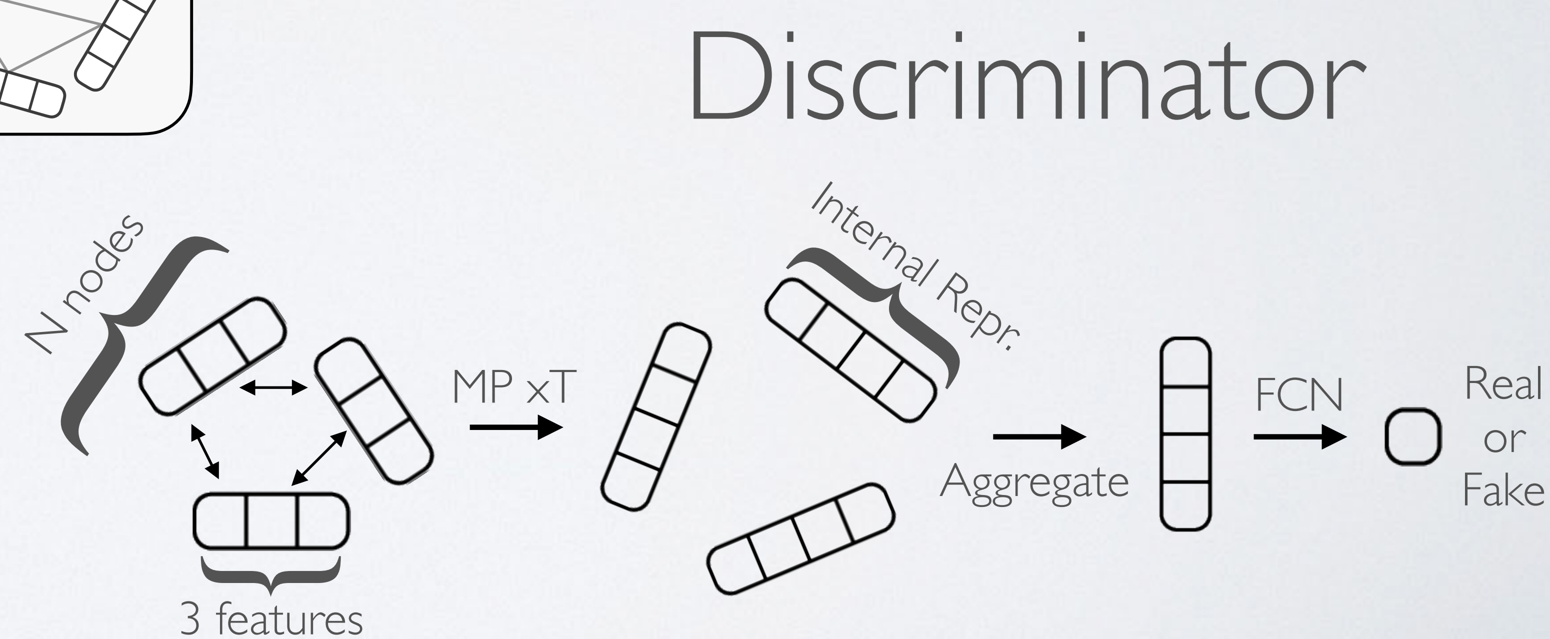
To allow for variable sized graphs we ‘mask’ extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$





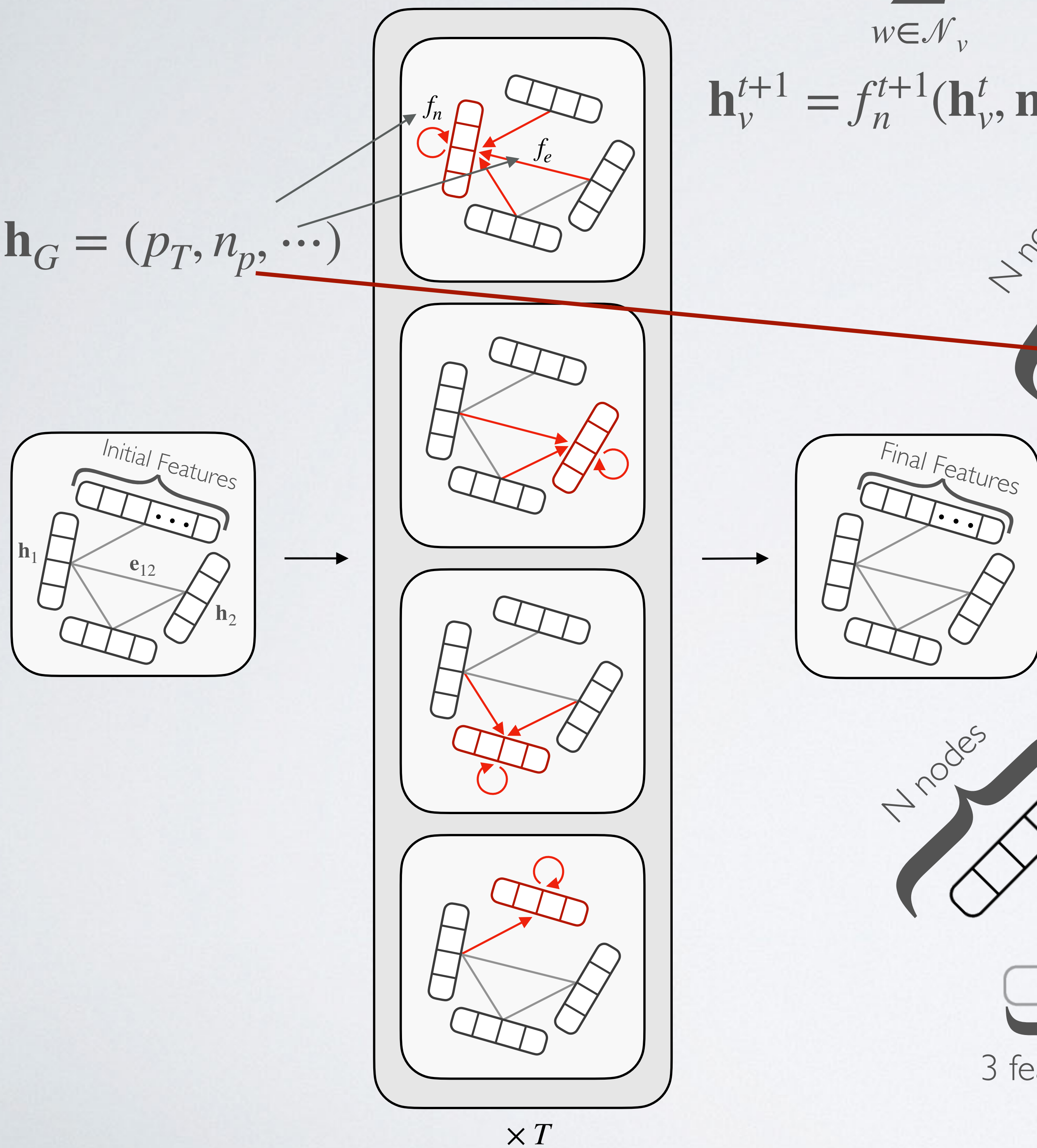


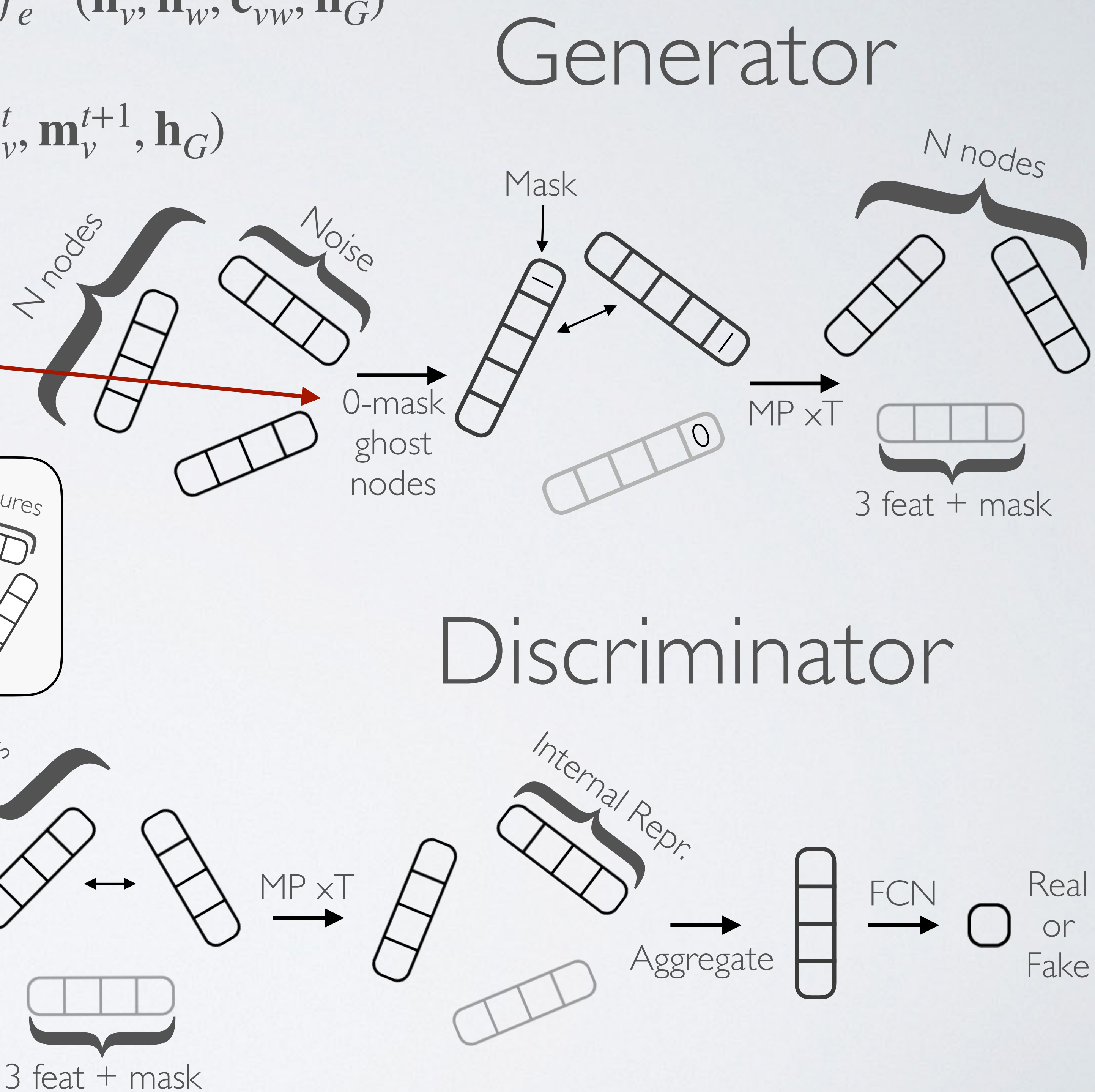
GAN ARCHITECTURE

To allow for variable sized graphs we 'mask' extra nodes with an additional binary feature:

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} f_e^{t+1}(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t, \mathbf{h}_G)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}, \mathbf{h}_G)$$



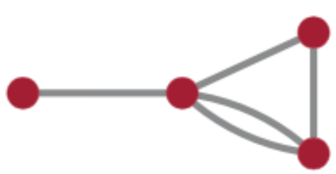


ENERGY FLOW POLYNOMIALS

$$z_i = \frac{p_{T,i}}{p_{T,J}}, \qquad p_{T,J} \equiv \sum_{i=1}^{M} p_{T,i},$$

$$\theta_{ij} = \left(\Delta y_{ij}^2 + \Delta\phi_{ij}^2\right)^{\beta/2},$$

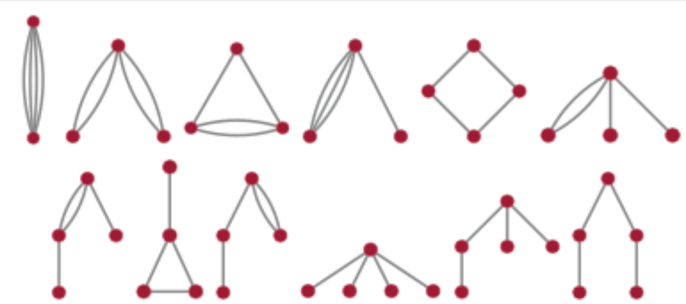
 $_j \iff \sum_{i_j=1}^{M} z_{i_j},$ $\qquad k$ $\ell \iff \theta_{i_k i_\ell}.$

 $= \sum_{i_1=1}^{M}\sum_{i_2=1}^{M}\sum_{i_3=1}^{M}\sum_{i_4=1}^{M} z_{i_1} z_{i_2} z_{i_3} z_{i_4} \theta_{i_1 i_2} \theta_{i_2 i_3} \theta_{i_2 i_4}^2 \theta_{i_3 i_4}.$

EFPs form a complete basis for IRC observables

e.g. (relative) jet mass:

$$\frac{m_J^2}{p_{TJ}^2} = \sum_{i_1=1}^{M}\sum_{i_2=1}^{M} z_{i_1} z_{i_2} (\cosh(\Delta y_{i_1 i_2}) - \cos(\Delta\phi_{i_1 i_2})) = \frac{1}{2} \times$$  $$+ \cdots . \quad (2.7)$$

Degree	Connected Multigraphs
$d = 0$	
$d = 1$	
$d = 2$	
$d = 3$	
$d = 4$	
$d = 5$