UNIVERSITÉ DE GENÈVE

SIP stochastic information processing

TRANSIT

your events into a new mass: Fast background interpolation for weakly-supervised anomaly searches

Ivan Oleksiyuk*, Tobias Golling, Slava Voloshynovskiy

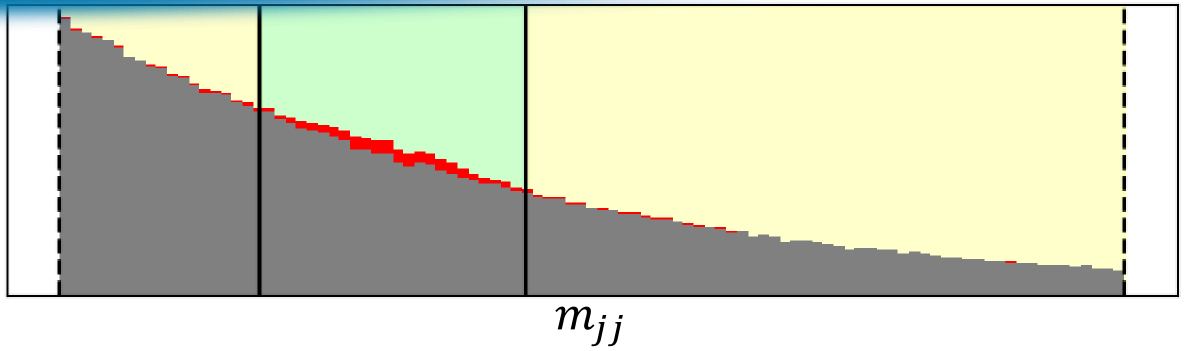
University of Geneva

*ivan.oleksiyuk@unige.ch

ML4Jets 2024 Paris

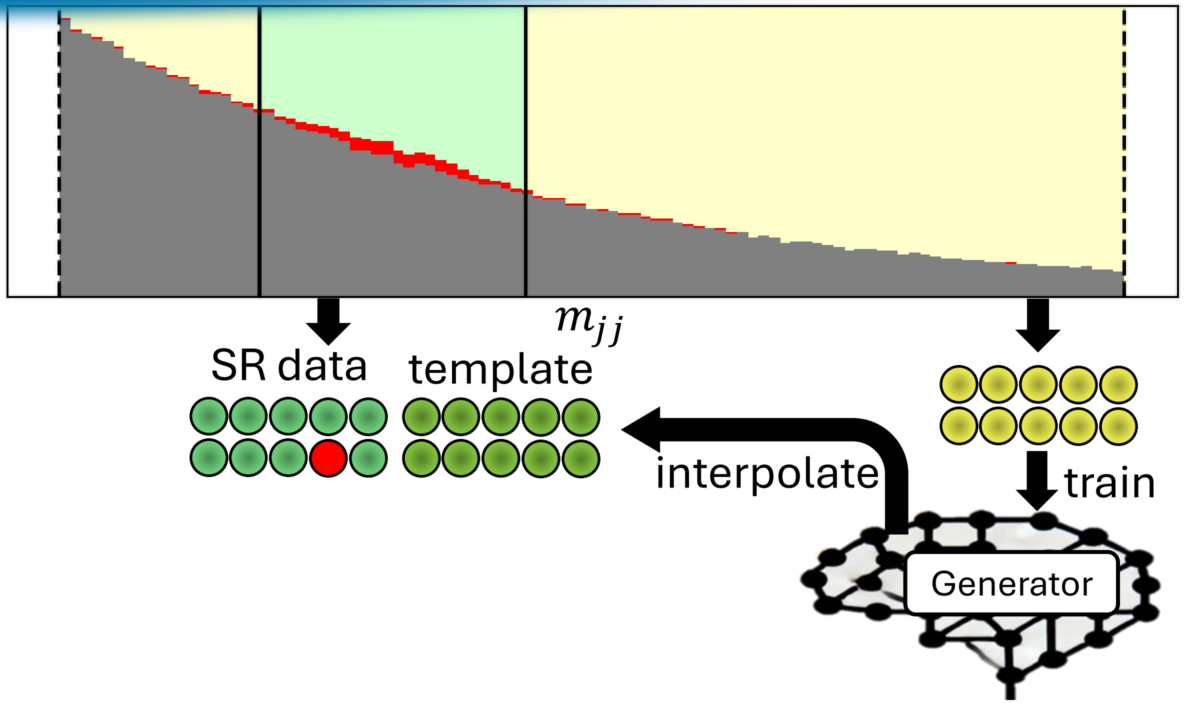
Data driven weakly-supervised searches

1. Select a signal region (SR) and sidebands (SB)



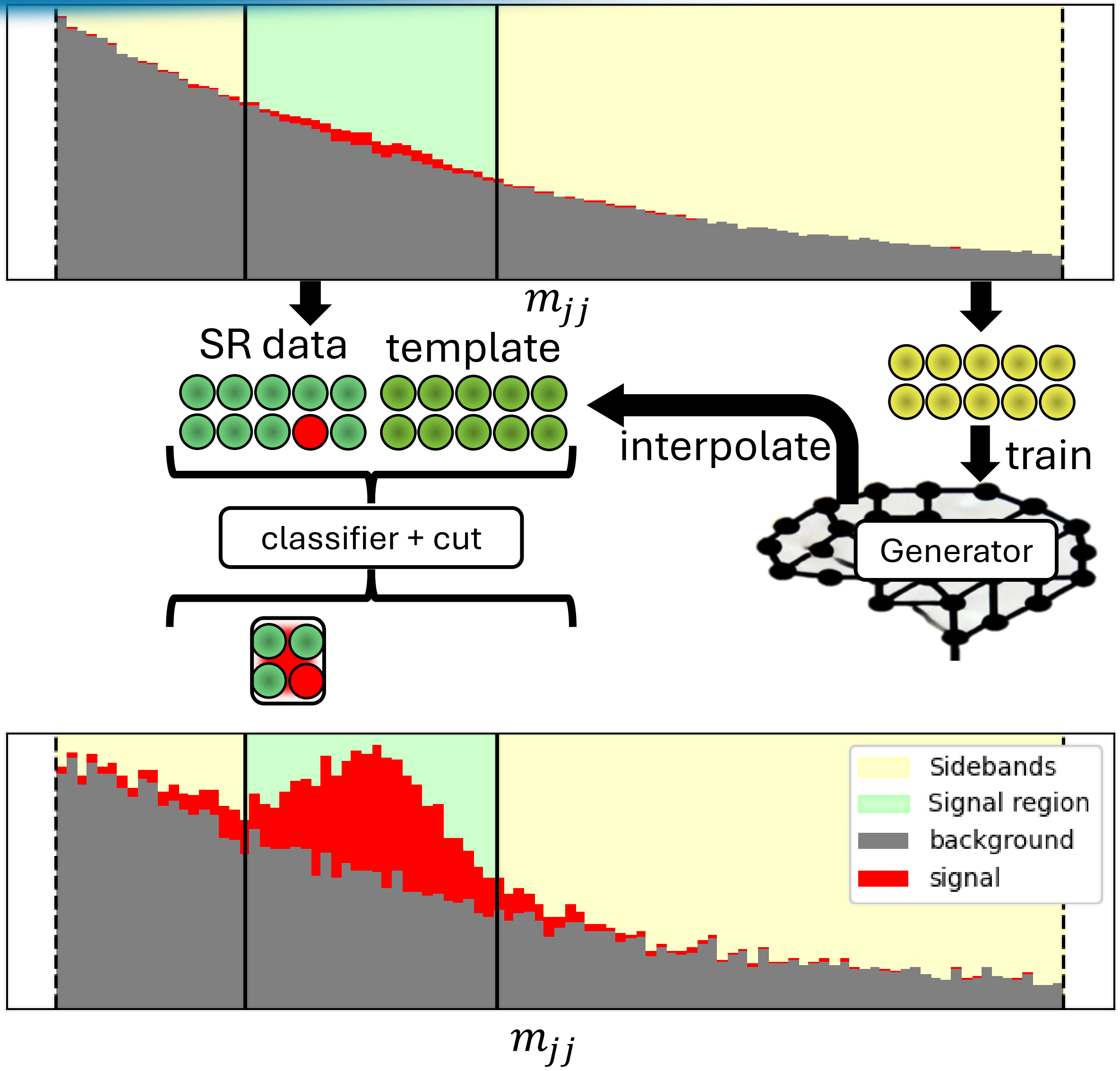
Data driven weakly-supervised searches

1. Select a signal region (SR) and sidebands (SB)
2. **Create a "template" that matches background in SR**



Data driven weakly-supervised searches

1. Select a signal region (SR) and sidebands (SB)
2. **Create a "template" that matches background in SR**
3. Use CWoLa on "template" and SR data to select the most anomalous data

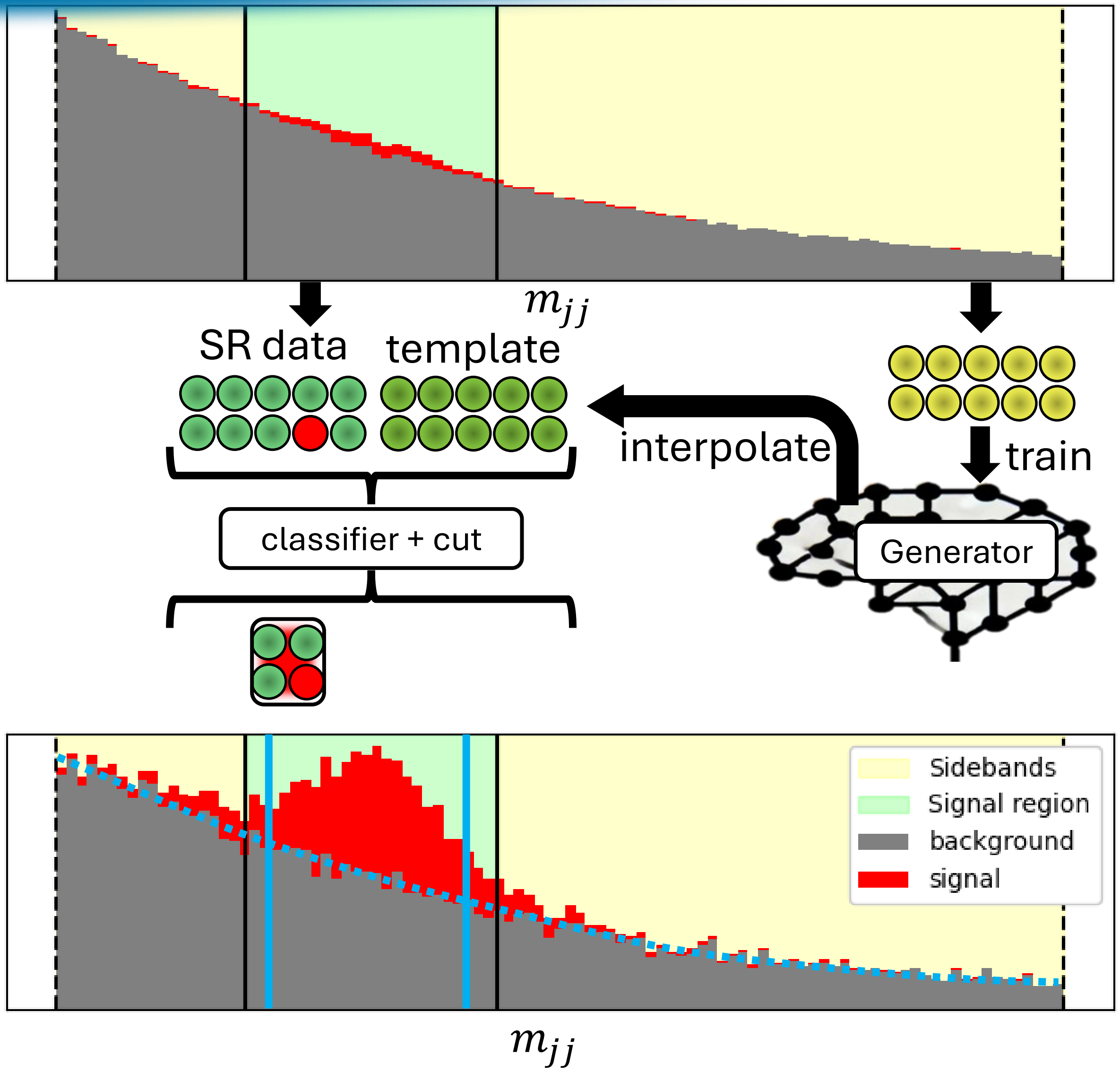


Data driven weakly-supervised searches

1. Select a signal region (SR) and sidebands (SB)
2. **Create a “template” that matches background in SR**
3. Use CWoLa on “template” and SR data to select the most anomalous data
4. Fit the background from sidebands, use Bump-Hunt to find excess, calculate significance or update limits

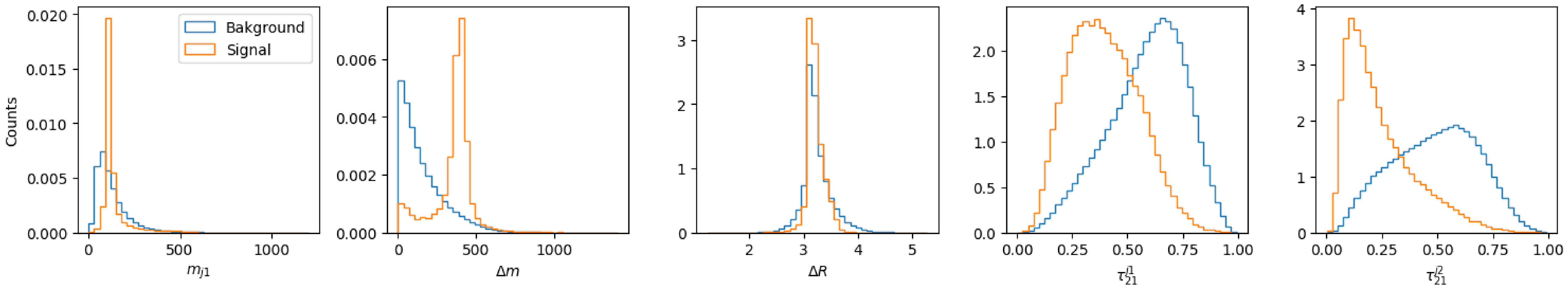
Significance Improvement (SI):

$$\frac{\text{Signal efficiency}}{\sqrt{\text{Background efficiency}}} = \frac{\varepsilon_{sig}}{\sqrt{\varepsilon_{bkg}}}$$



Benchmark data

- LHCO R&D dataset - the most popular benchmark (arXiv:2101.08320).
  - 1M QCD dijet events (background)
  - 100K $Z'(3.5TeV) \rightarrow X(500GeV)Y(100Gev) \rightarrow (qq)(qq)$
  - Signal has a prominent two prong structure
- Use widely accepted high level variables
  - m_{JJ} as resonant variable
  - $m_{J1},\ \Delta m_J = m_{J1} - m_{J2}, \tau_{12}^{J1}, \tau_{12}^{J2},\ \Delta R_{JJ} = \sqrt{\Delta\eta^2 + \Delta\varphi^2}$



Data driven weakly-supervised searches

Original CWoLa (arXiv:1902.02634) – using sidebands, as a crude template approximation

ML based template – Slow but good template

- SALAD (arXiv: 2001.05001) – classifier (MC based)
- FETA (arXiv: 2212.11285) – normalising flows (MC based)
- CATHODE (arXiv:2109.00546v3, arXiv: 2210.14924) – normalising flows
- CURTAINs (arXiv:2203.09470v3, arXiv: 2305.04646) – normalising flows
- DRAPES (arXiv:2312.10130) – diffusion
- Full Phase Space Resonant Anomaly Detection (arXiv: 2310.06897) – diffusion and flow matching

etc…

Non-ML template – Fast but limited quality

- RAD-OT (arXiv:2407.19818) – optimal transport

Data driven weakly-supervised searches

Original CWoLa (arXiv:1902.02634) – using sidebands, as a crude template approximation

ML based template – Slow but good template

- SALAD (arXiv: 2001.05001) – classifier (MC based)
- FETA (arXiv: 2212.11285) – normalising flows (MC based)
- CATHODE (arXiv:2109.00546v3, arXiv: 2210.14924) – normalising flows
- CURTAINs (arXiv:2203.09470v3, arXiv: 2305.04646) – normalising flows
- DRAPES (arXiv:2312.10130) – diffusion
- Full Phase Space Resonant Anomaly Detection (arXiv: 2310.06897) – diffusion and flow matching

etc…

Non-ML template – Fast but limited quality

- RAD-OT (arXiv:2407.19818) – optimal transport

Good template with speedup (just recent)

- CURTAINsF4F (arXiv: 2305.04646)
- SIGMA (arXiv2410.20537) – see Ranits talk
- **TRANSIT - this talk!**

Why do we want a fast method?

The algorithm on the last page seems simple…
but we have to repeat it:

- $\times\, O(10)$ signal regions
- $\times\, O(3)$ different variable combinations
- $\times\, O(20)$ signal models for limit setting
- $\times\, O(10)$ signal injection values for limit setting
- $\times\, O(20)$ validation datasets
- etc…

$\boldsymbol{O(10000)}$ template generation trainings per analysis

In many cases even more:

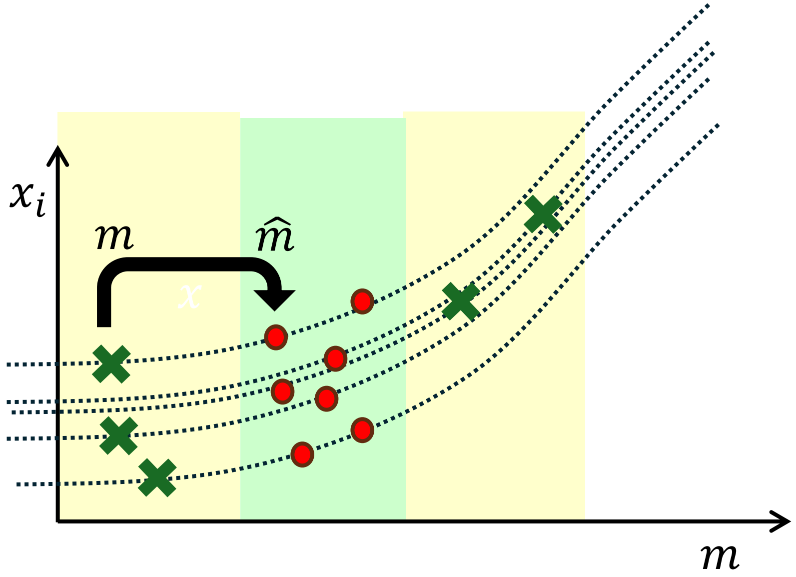
- $O(1000)$ times to get a distribution of test statistic
- $O(100)$ patches stellar streams in SkyCURTAINS (Stevens talk yesterday)

Main Idea

- Generative models learn the whole joint probability distribution $p(x_1, \dots, x_n, m)$ that include all complicated correlations between $x_1, \dots, x_n$.

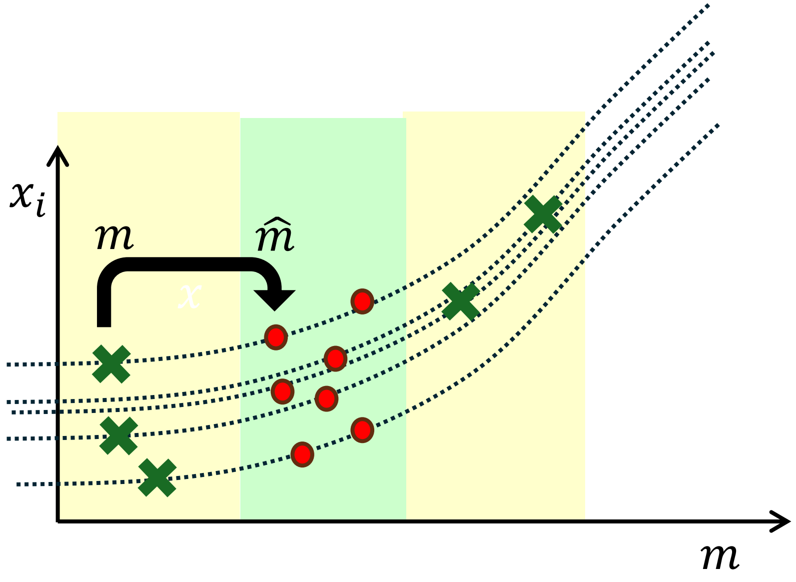
Main Idea

- Generative models learn the whole joint probability distribution $p(x_1, \dots, x_n, m)$ that include all complicated correlations between $x_1, \dots, x_n$.
- Why not morph $\boldsymbol{x} \sim p(\boldsymbol{x}|m)$ such that they match different mass $\hat{\boldsymbol{x}} = f_\theta(\boldsymbol{x}|m, \hat{m}) \sim p(\hat{\boldsymbol{x}}|\hat{m})$ instead?
- No need to learn correlations between $x_1, \dots, x_n$ if they do not change with m, e.g. if $p(x_k, m) = p(x_k)p(m)$ NN just learns identity!



Main Idea

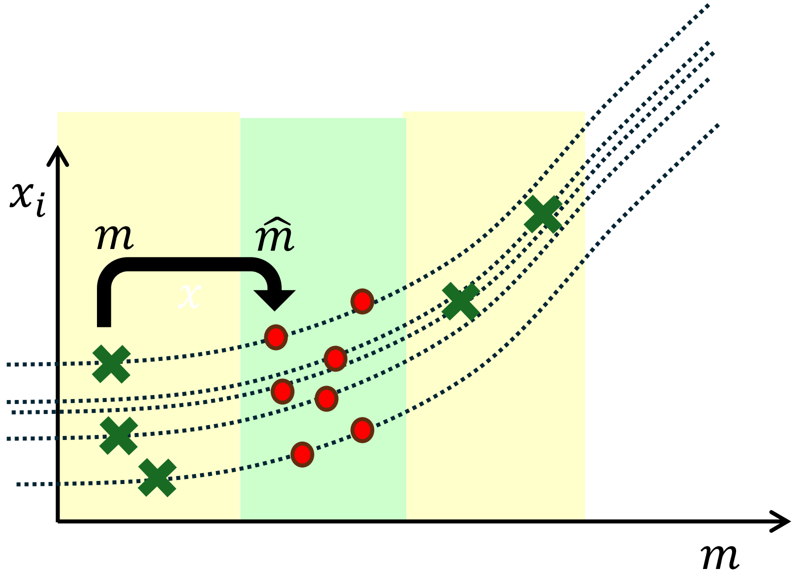
- Generative models learn the whole joint probability distribution $p(x_1, \ldots, x_n, m)$ that include all <u>complicated</u> correlations between $x_1, \ldots, x_n$.
- Why not morph $\boldsymbol{x} \sim p(\boldsymbol{x}|m)$ such that they match different mass $\hat{\boldsymbol{x}} = f_\theta(\boldsymbol{x}|m, \hat{m}) \sim p(\hat{\boldsymbol{x}}|\hat{m})$ instead?
- No need to learn correlations between $x_1, \ldots, x_n$ if they do not change with m, e.g. if $p(x_k, m) = p(x_k)p(m)$ NN just learns identity!
- CURTAINs/CURTAINsF4F already do similar, but…
  <u>Flow/INN training is hard!</u>
- Can we train a simpler model?



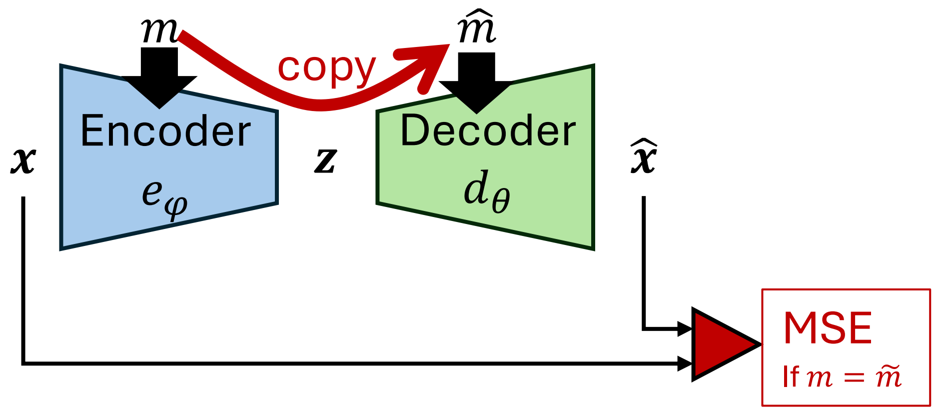
Main Idea

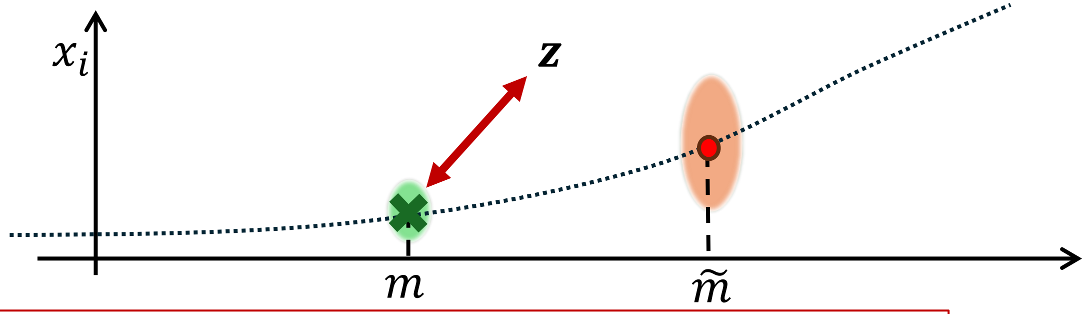
- Generative models learn the whole joint probability distribution $p(x_1, \ldots, x_n, m)$ that include all <u>complicated</u> correlations between $x_1, \ldots, x_n$.
- Why not morph $\boldsymbol{x} \sim p(\boldsymbol{x}|m)$ such that they match different mass $\widehat{\boldsymbol{x}} = f_\theta(\boldsymbol{x}|m, \widehat{m}) \sim p(\widehat{\boldsymbol{x}}|\widehat{m})$ instead?
- No need to learn correlations between $x_1, \ldots, x_n$ if they do not change with m, e.g. if $p(x_k, m) = p(x_k)p(m)$ NN just learns identity!
- CURTAINs/CURTAINsF4F already do similar, but… <u>Flow/INN training is hard!</u>
- Can we train a simpler model?
- Use **TRANSIT**

(**TR**ansport **A**dversarial **N**etwork for **S**mooth **I**n**T**erpolarion)!



Model losses

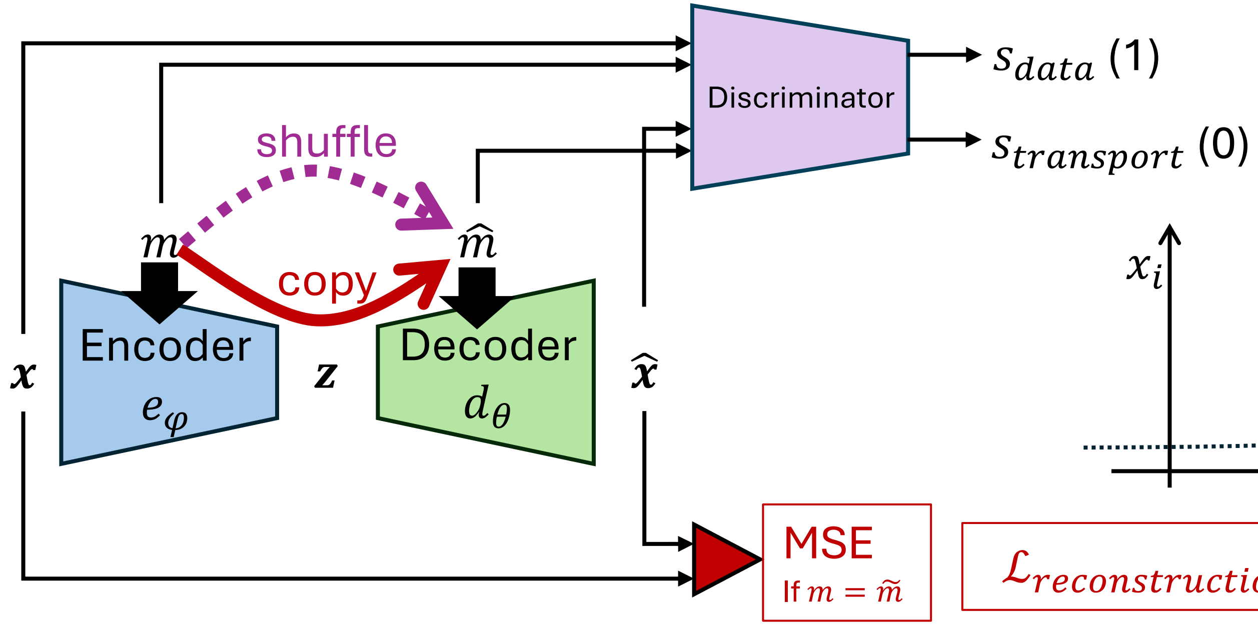




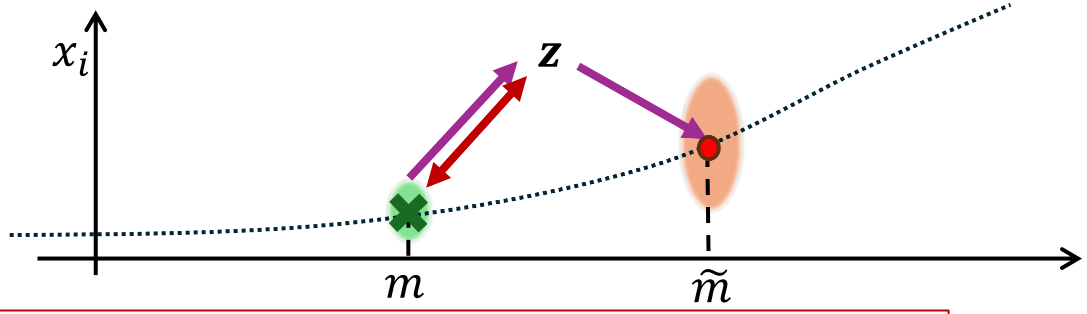
$$\mathcal{L}_{reconstruction} = \mathbb{E}_{p(x,m)}[\|d_\theta(e_\varphi(\boldsymbol{x}, m), m) - \boldsymbol{x}\|]$$

- $\mathcal{L}_{reconstruction}$ - ensures that no change is done if the mass is the same

Model losses



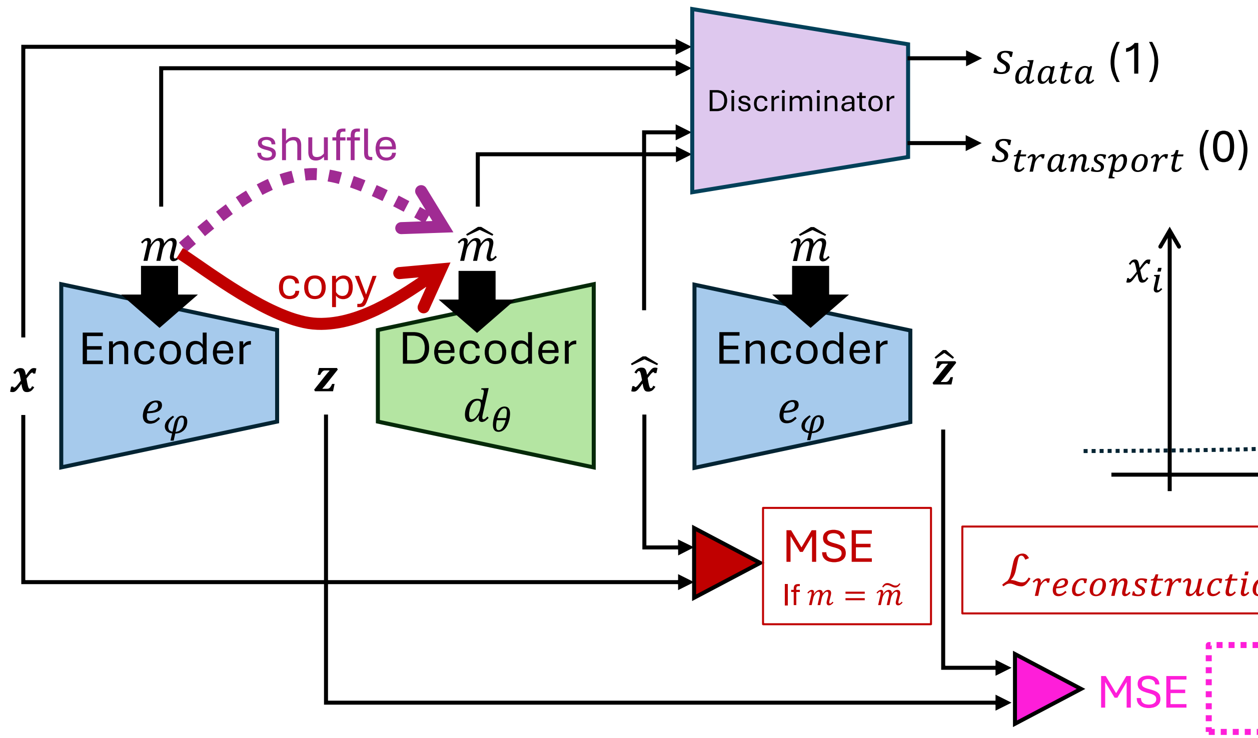
$$\mathcal{L}_{transport} = -\mathcal{L}_{discriminator} = -BCE =$$
$$=0.5\log(s_{data}) + 0.5\log(1 - s_{transport})$$
$$\Leftrightarrow \min(D_{KL}(p(\boldsymbol{x}|m)||\hat{p}_{\theta,\varphi}(\hat{\boldsymbol{x}}|\hat{m})))$$



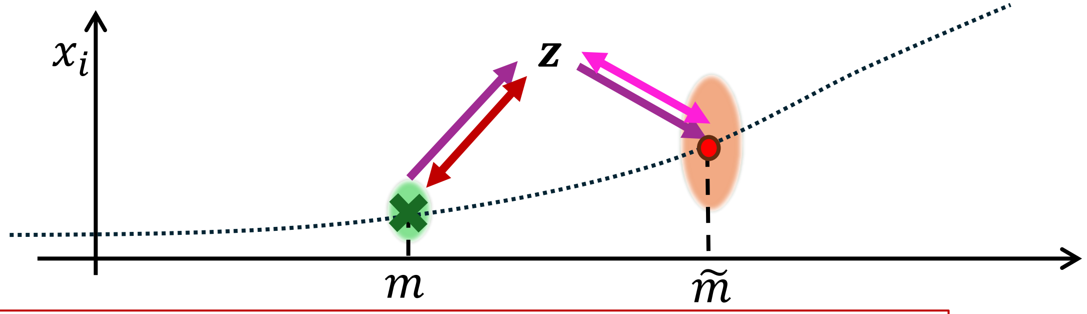
$$\mathcal{L}_{reconstruction} = \mathbb{E}_{p(x,m)}[\|d_\theta(e_\varphi(\boldsymbol{x}, m), m) - \boldsymbol{x}\|]$$

- $\mathcal{L}_{transport}$ - ensures that the generated conditional distribution = distribution of the data
- $\mathcal{L}_{reconstruction}$ - ensures that no change is done if the mass is the same

Model losses



$$\mathcal{L}_{transport} = -\mathcal{L}_{discriminator} = -BCE = $$
$$=0.5\log(s_{data}) + 0.5\log(1 - s_{transport})$$
$$\Leftrightarrow \min\left(D_{KL}\left(p(\boldsymbol{x}|m)||\hat{p}_{\theta,\varphi}(\hat{\boldsymbol{x}}|\hat{m})\right)\right)$$

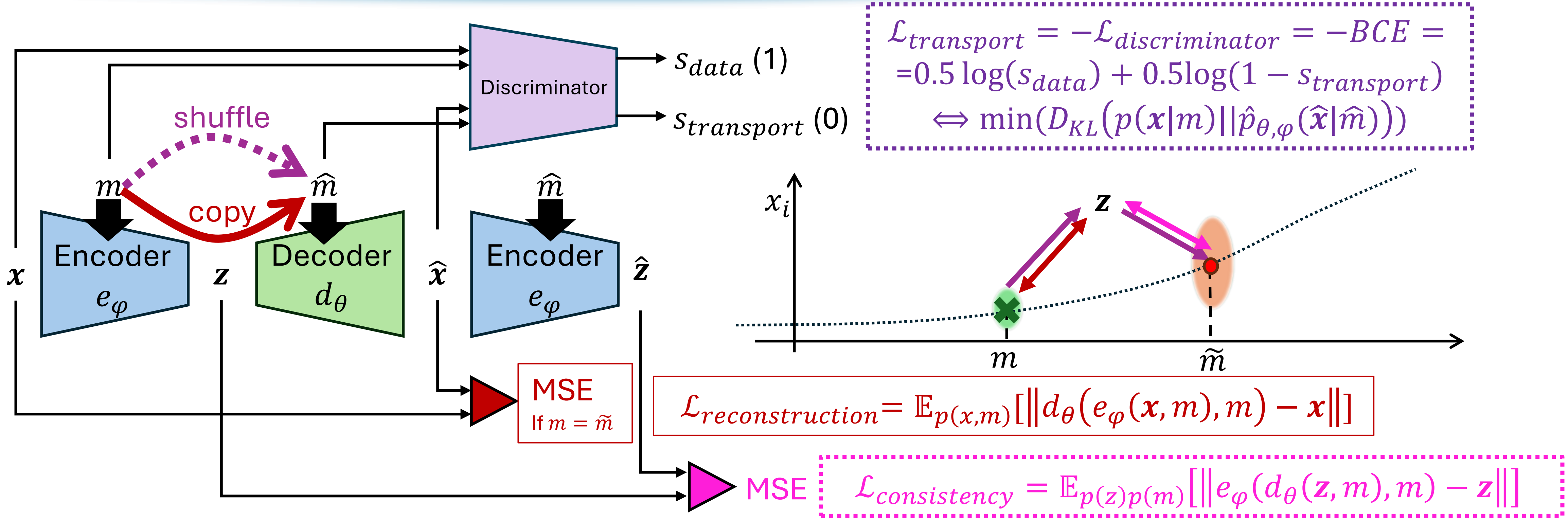


$$\mathcal{L}_{reconstruction} = \mathbb{E}_{p(x,m)}\left[\left\|d_\theta\left(e_\varphi(\boldsymbol{x},m),m\right) - \boldsymbol{x}\right\|\right]$$

$$\mathcal{L}_{consistency} = \mathbb{E}_{p(z)p(m)}\left[\left\|e_\varphi(d_\theta(\boldsymbol{z},m),m) - \boldsymbol{z}\right\|\right]$$

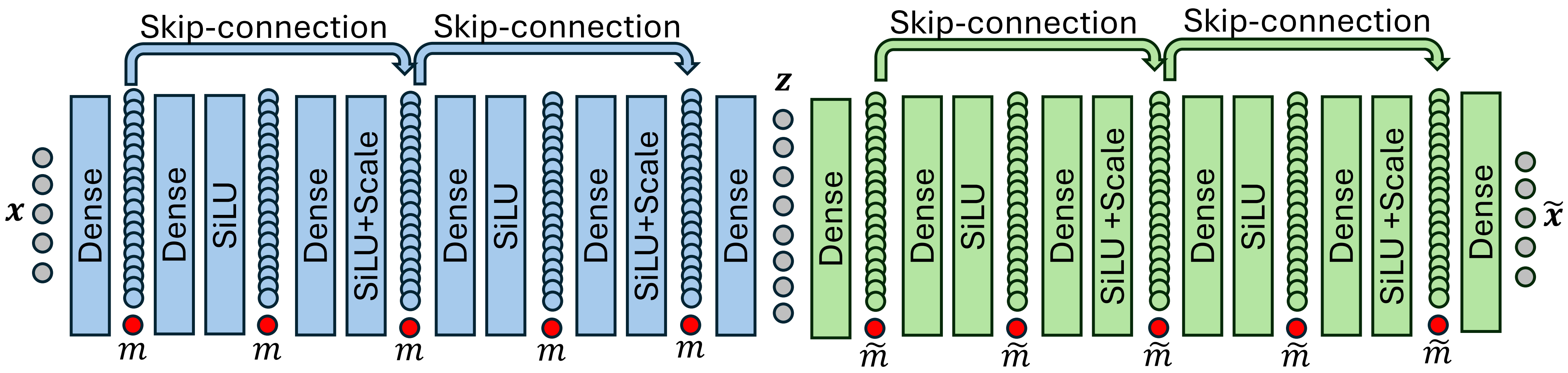
- $\mathcal{L}_{transport}$ - ensures that the generated conditional distribution = distribution of the data
- $\mathcal{L}_{reconstruction}$ - ensures that no change is done if the mass is the same
- $\mathcal{L}_{consistency}$ - make latent representations of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ equal for any $\hat{m}$ so that one can unambiguously return from $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}$

Towards mass decorrelation



- If $\mathcal{L}_{transport}$ achieves minimum it means $(\hat{\boldsymbol{x}}, \hat{m})$ are fully paired
  m is shuffled relative to $\hat{m} \Rightarrow \hat{m} \perp m,\ \hat{\boldsymbol{x}} \perp m$
- $\Rightarrow\ \hat{\boldsymbol{z}} = e_\varphi(\hat{\boldsymbol{x}}, \hat{m}) \perp m$
- If $\mathcal{L}_{consistency}$ is saturated $\Leftrightarrow \boldsymbol{z} \approx \hat{\boldsymbol{z}} \Longrightarrow\ \boldsymbol{z} \perp m$

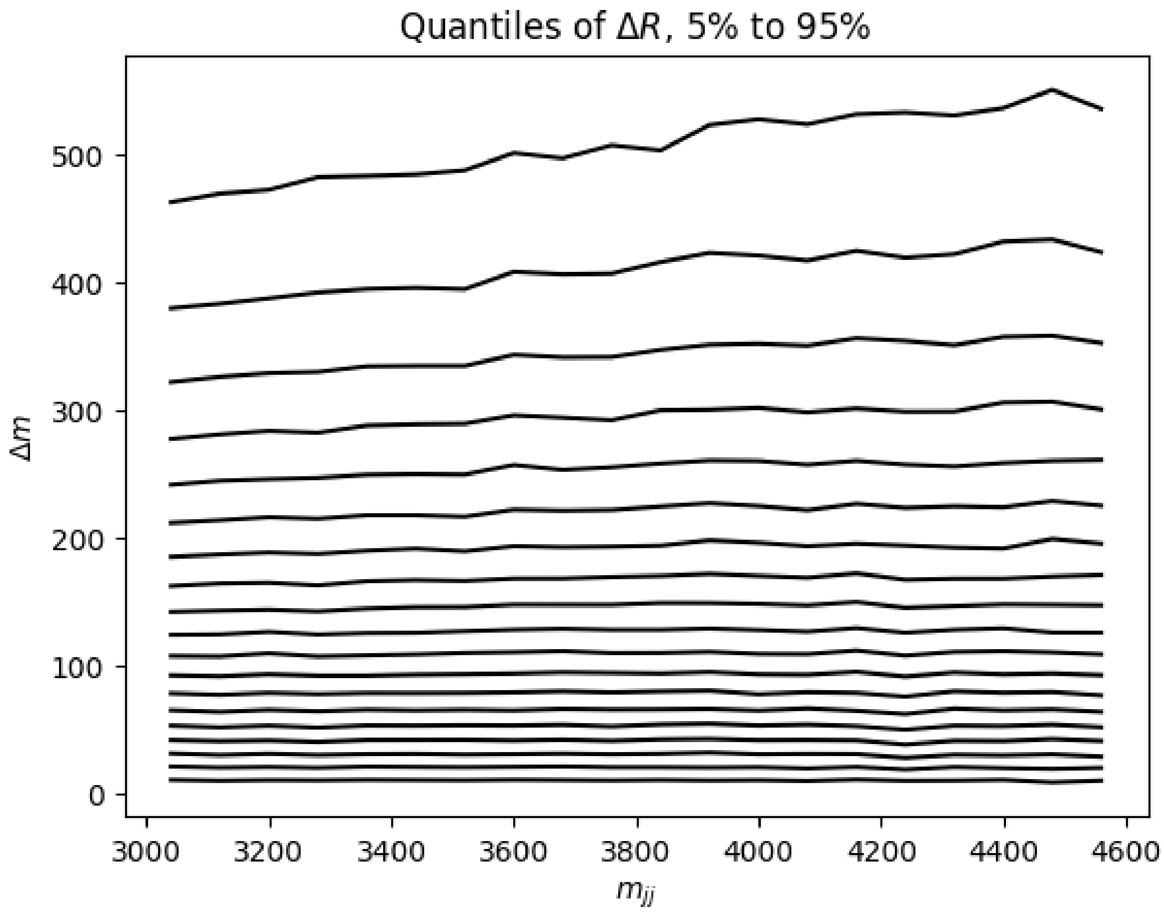
Architecture

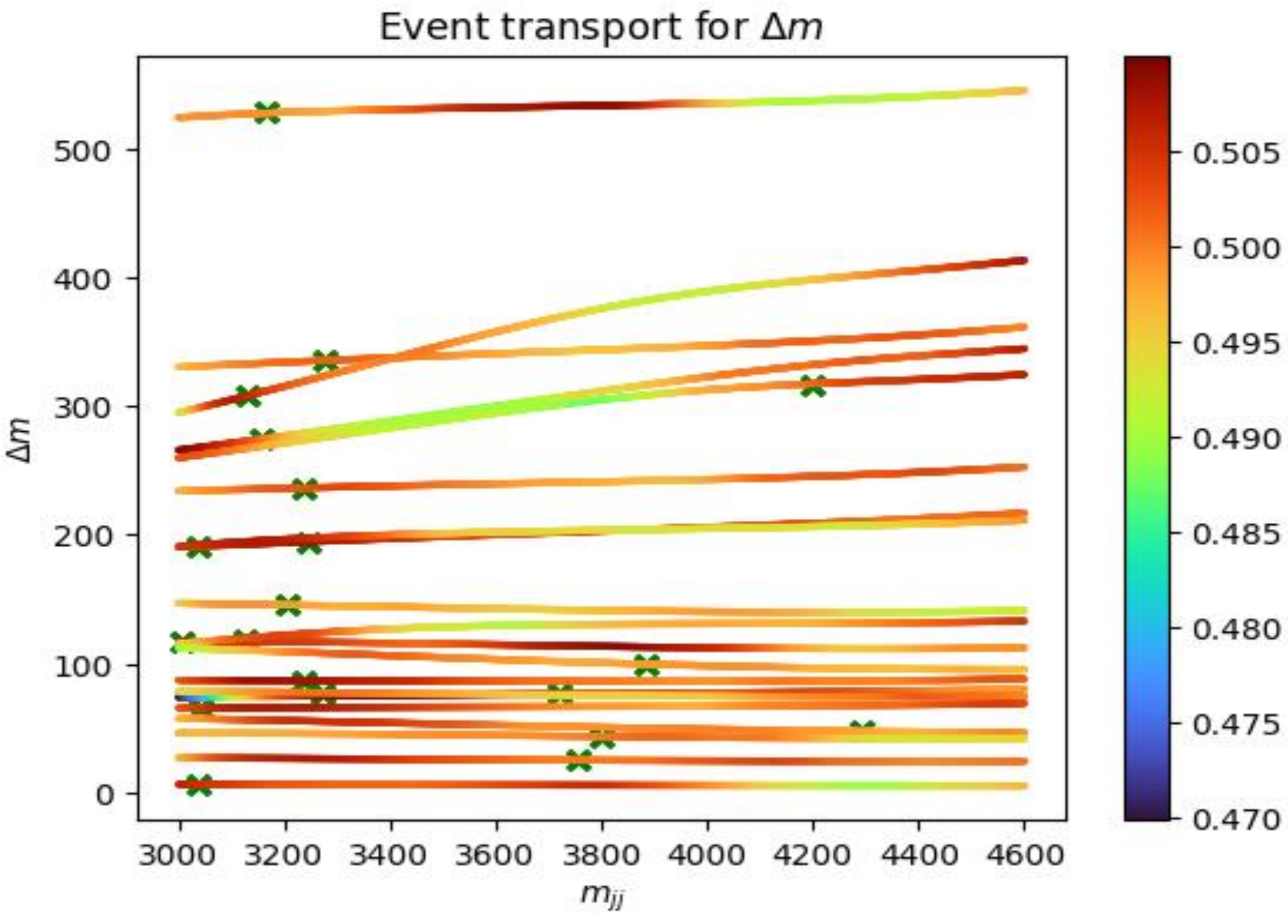


- Smoothness achieved by using SiLU activation
- Latent space is not restrictive as $\dim(\boldsymbol{z}) > \dim(\boldsymbol{x})$
- Conditioning used in every layer
- Identity is easily learnable by making an identity in dense layers and all scales 0
- Discriminator - MLP with 4 hidden layers of width 64 and Silu activations

Trajectories: low correlation Δm_J

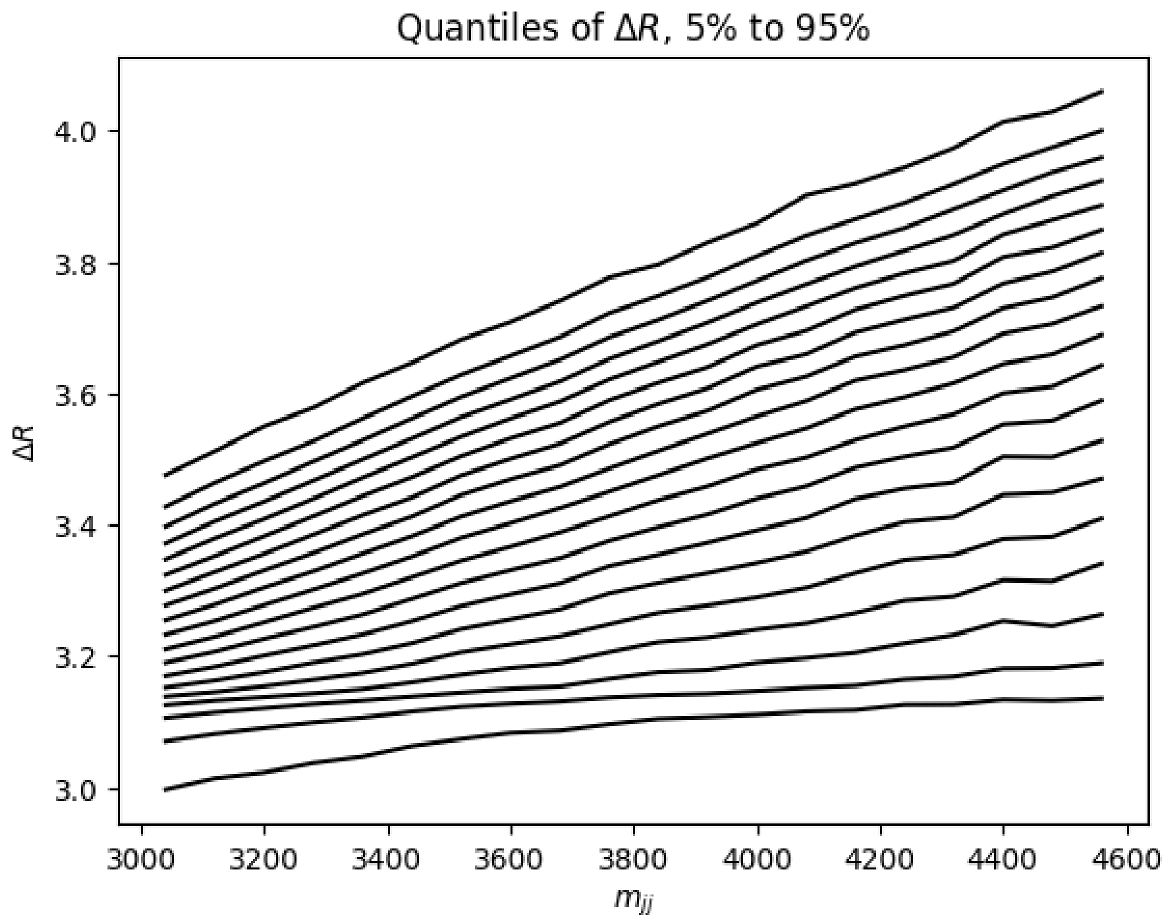
For uncorrelated variables like Δm_J the trajectories are mostly horizontal lines, however they are curved in the tails of the distribution to account for some m_{JJ} dependance

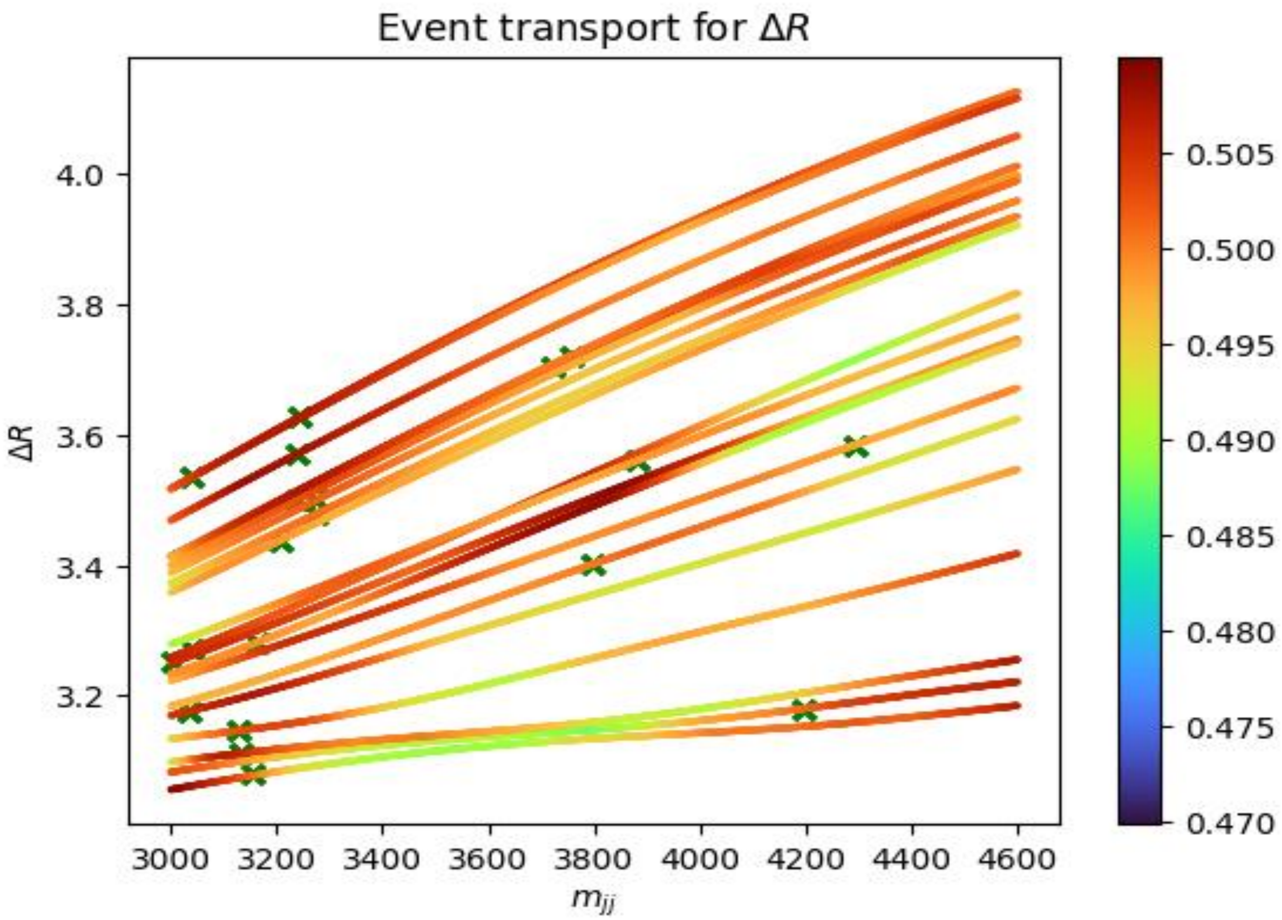




Trajectories: high correlation ΔR_{JJ}

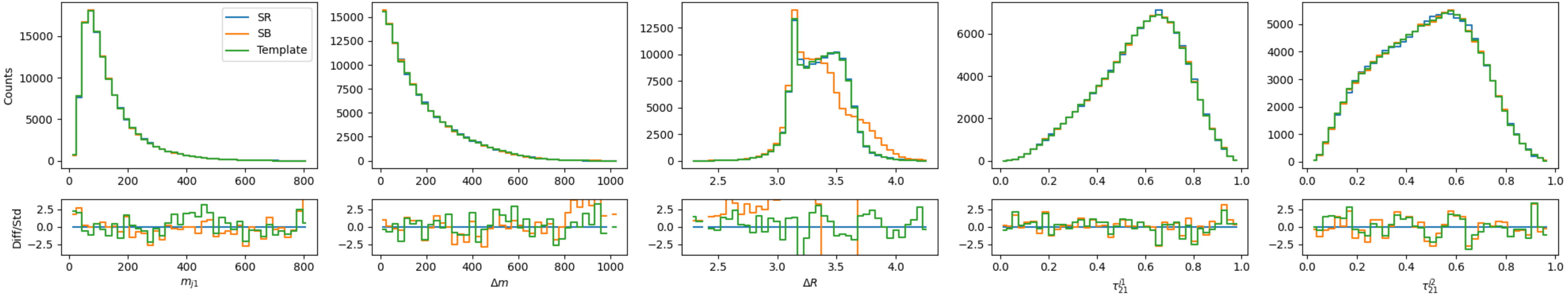
For ΔR_{JJ} the trajectories closely follow the quantiles of the distribution thus ensuring the correct distribution morph





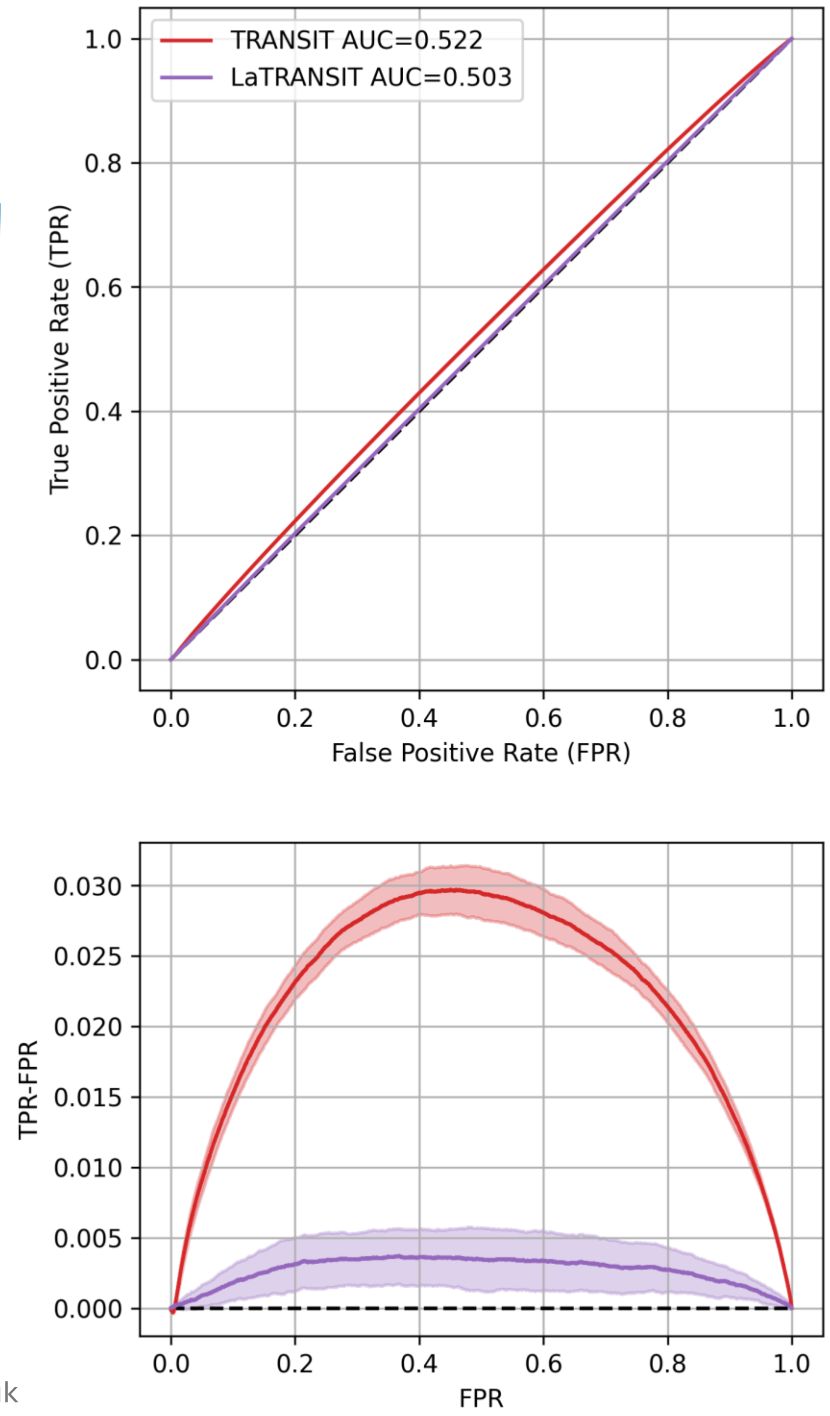
Template

- Transport event from SB into $\tilde{m}$ sampled from SR
- The template matches well in all variables, while sidebands had a significantly different distribution
- We can even resolve the “spike” feature on ΔR_{JJ}



CWoLa

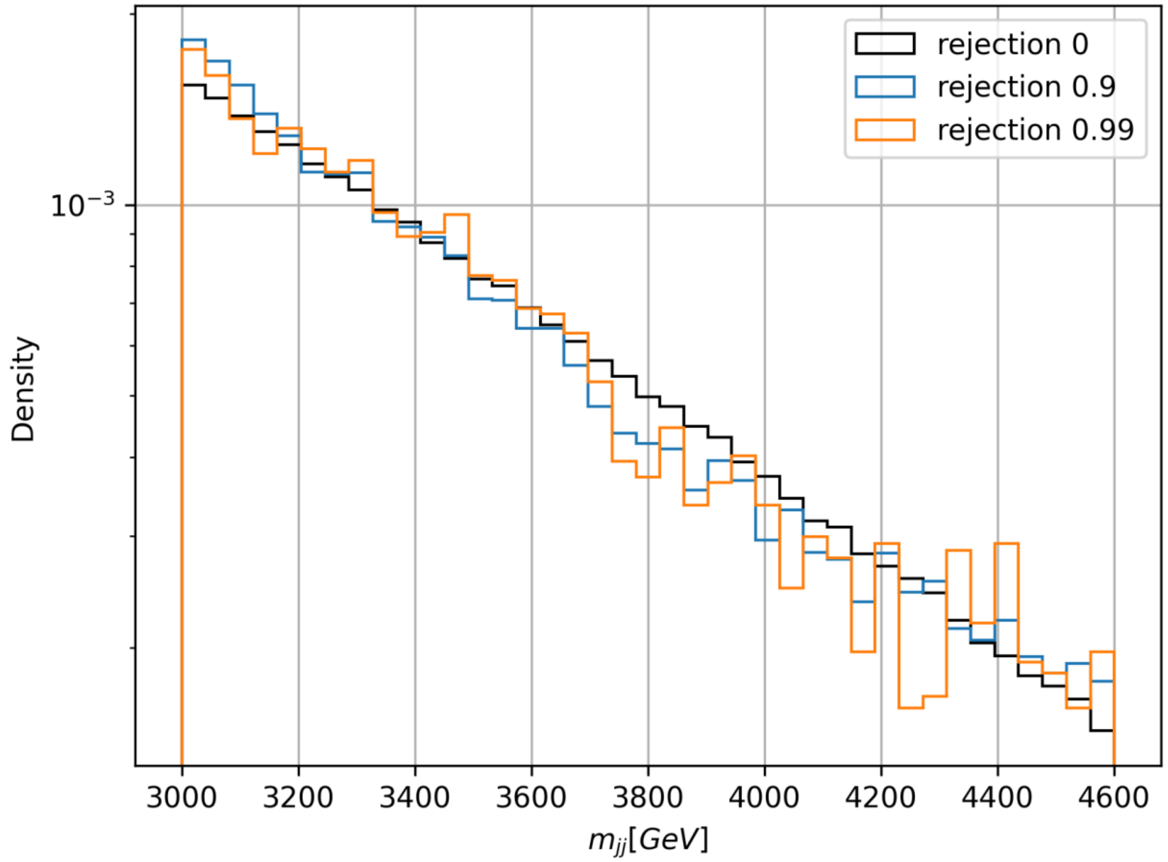
- Boosted Decision Trees
  - Fast
  - Robust to noise variables
- Ensemble of 5 classifiers
- 5-fold cross-validation
- Classify TRANSIT template vs SR data
- Classify latent space representations of SB vs latent space representations of SR for LaTRANSIT (analogy with LaCATHODE)



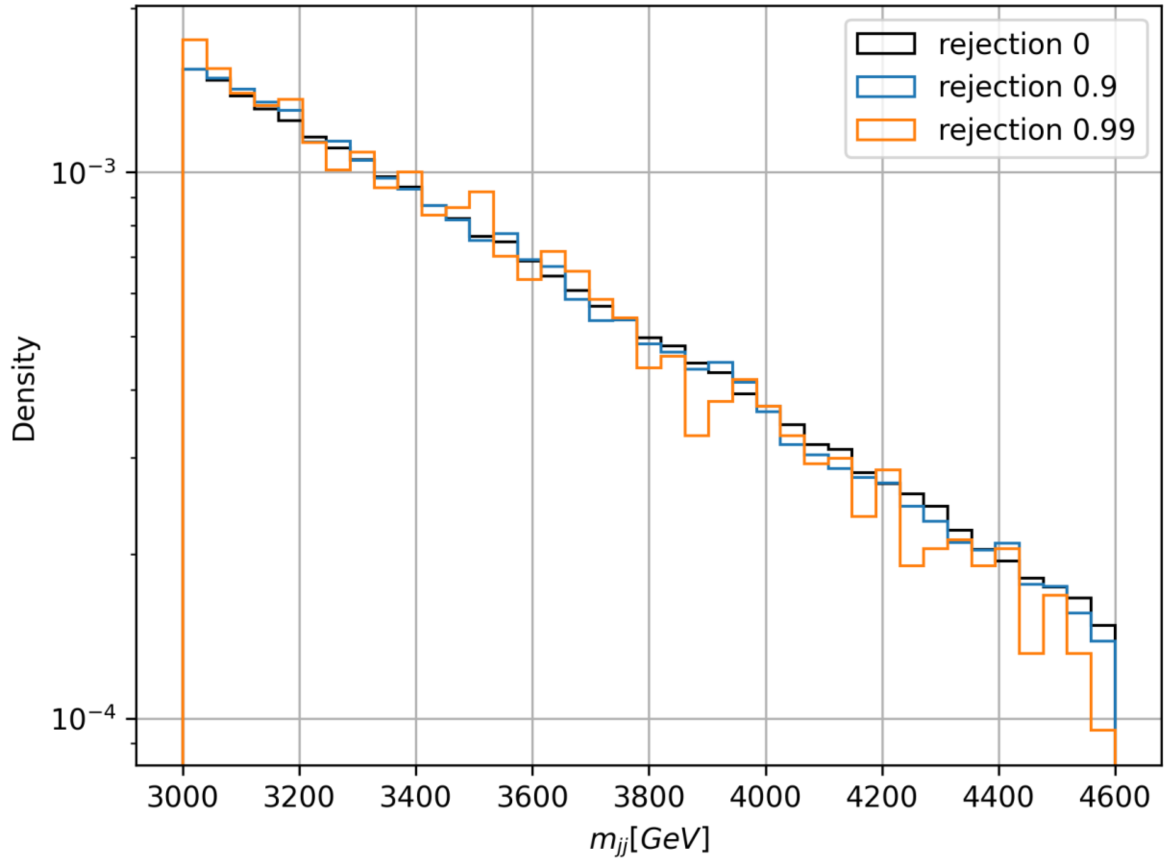
Mass Sculpting

- Select only background using CWoLa score threshold.
- Latent features have practically no correlation

$m_{J1}, \Delta m_J \tau_{12}^{J1}, \tau_{12}^{J2}, \Delta R_{JJ}$



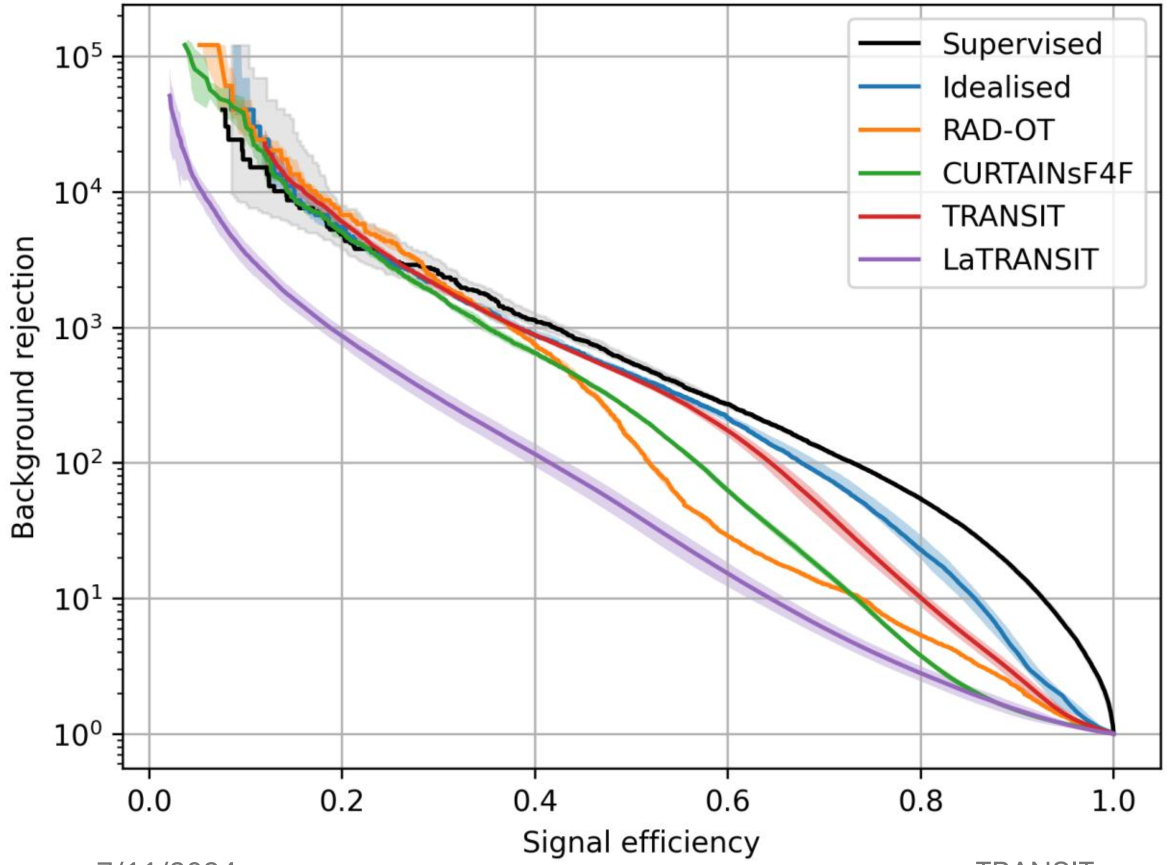
Latent space variables

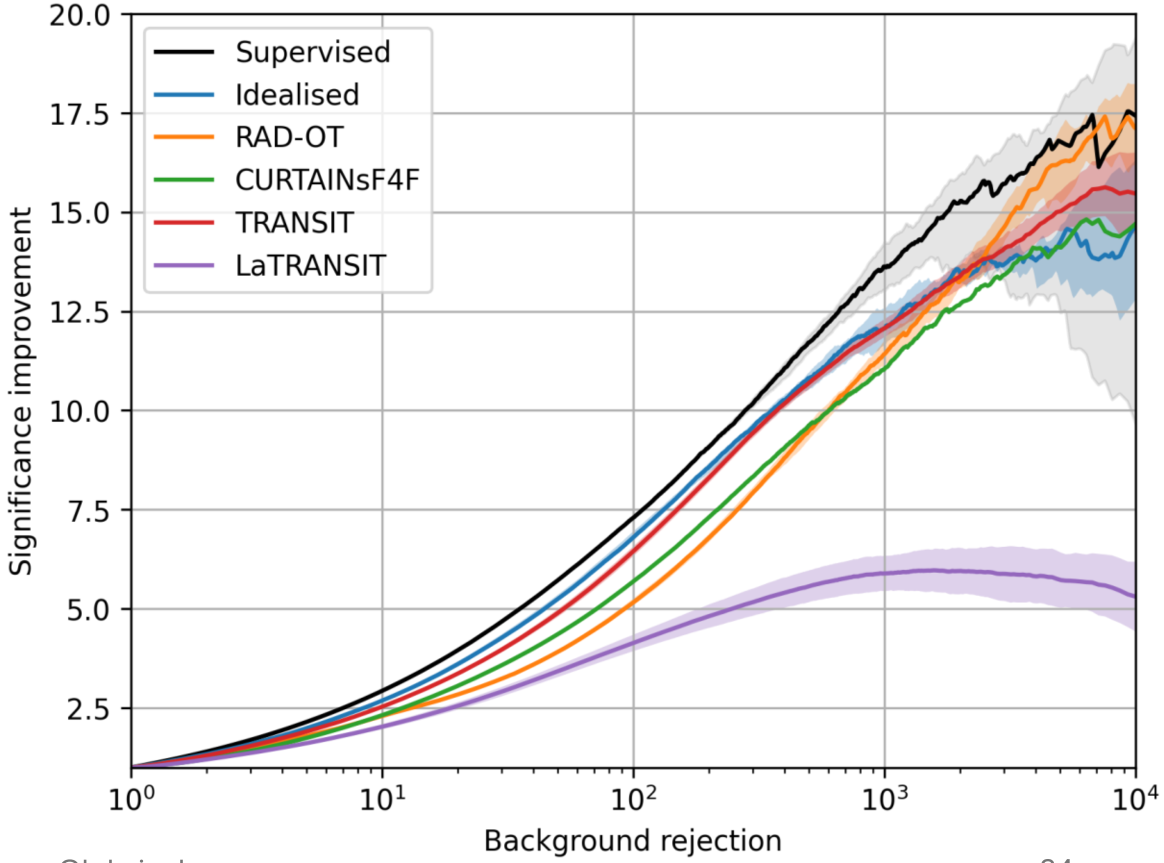


Significance improvement

- Inject 3000 signal which corresponds to 0.3% signal contamination in the full data
- Here TRANSIT performs better then both RAD-OT and CURTAINsF4F* and is close to Idealised case

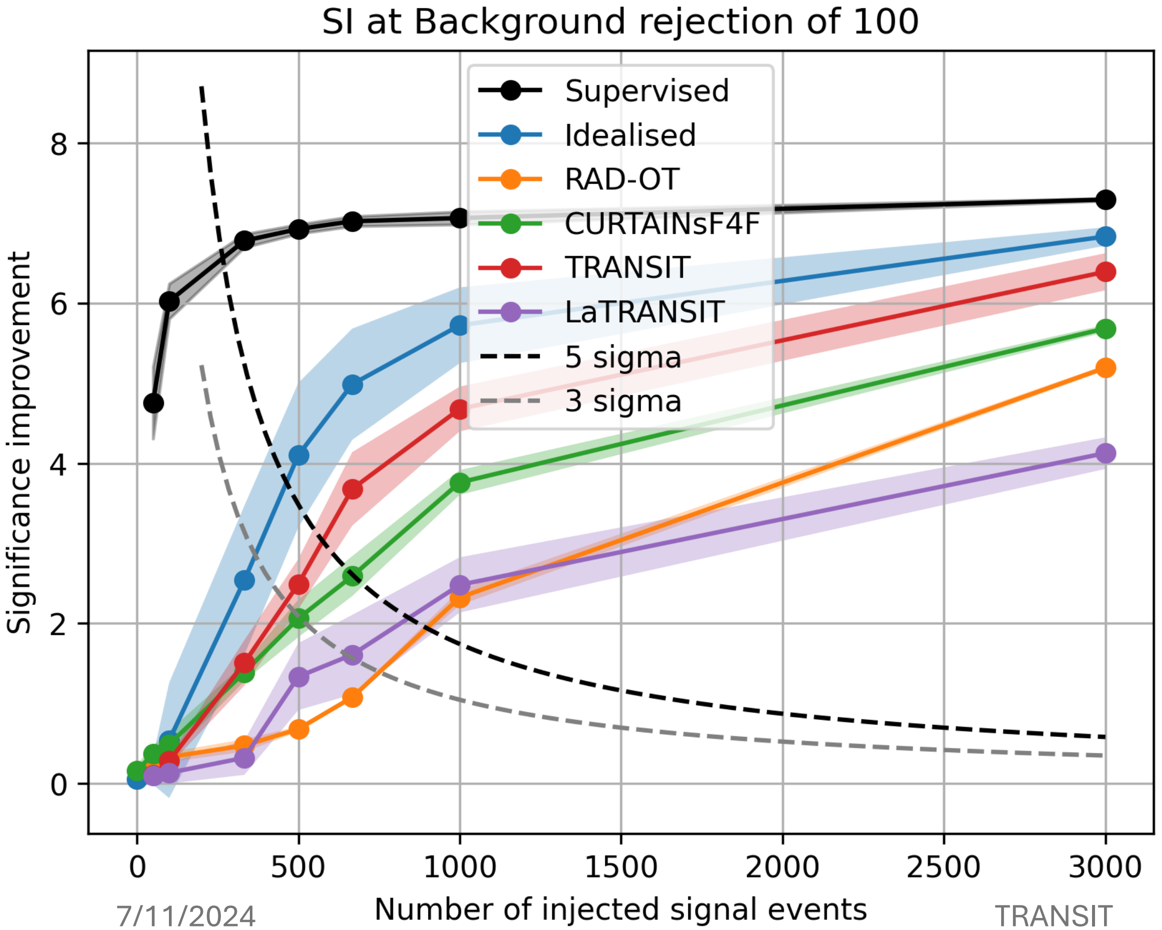
*Results for other methods are provided by authors of ArXiv: 2305.04646 and ArXiv: 2407.19818

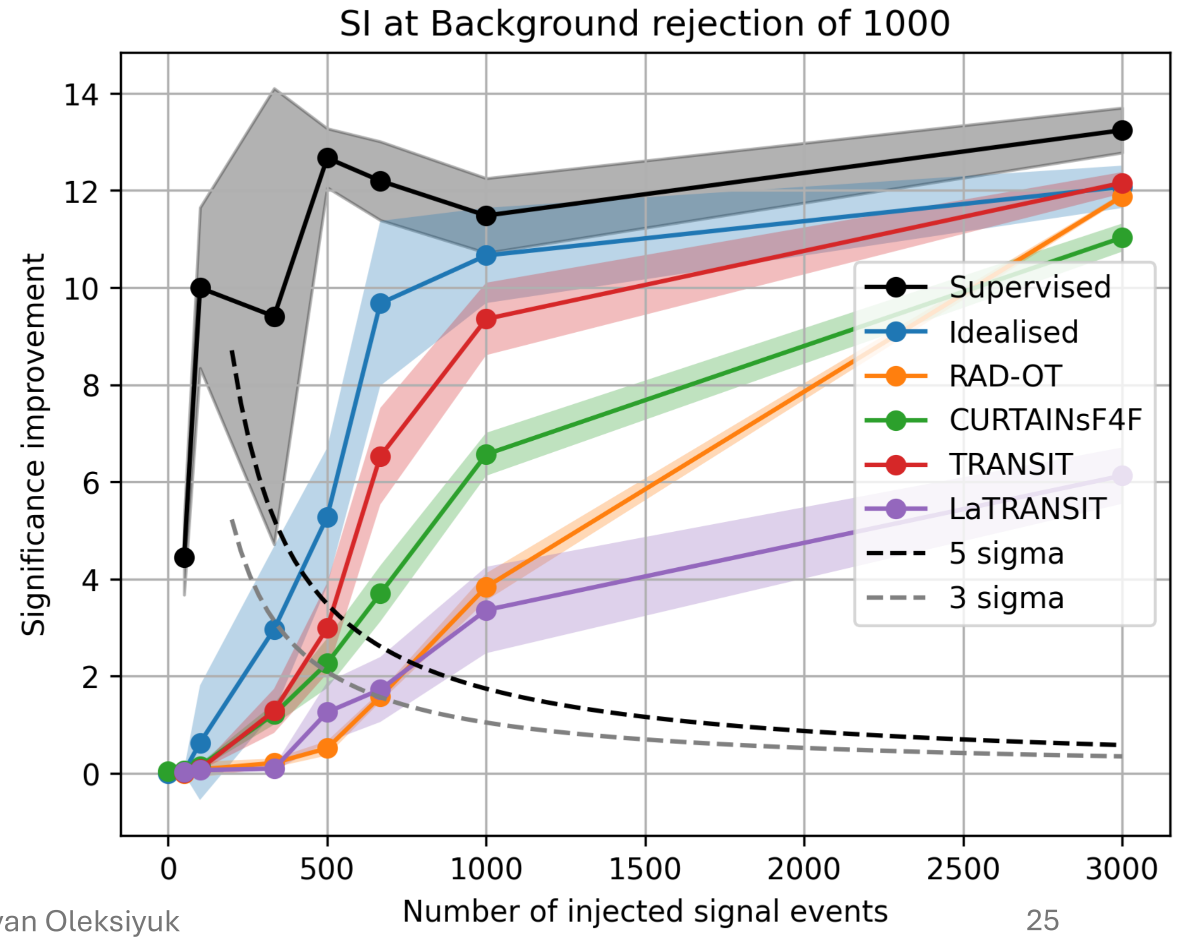




Significance improvement

- Dashed curves: SI needed for 3σ or 5σ excess in a 1 bin counting experiment with perfectly estimated background N from sidebands.
- In this region TRANSIT is outperforms both CurtainsF4F and RAD-OT*

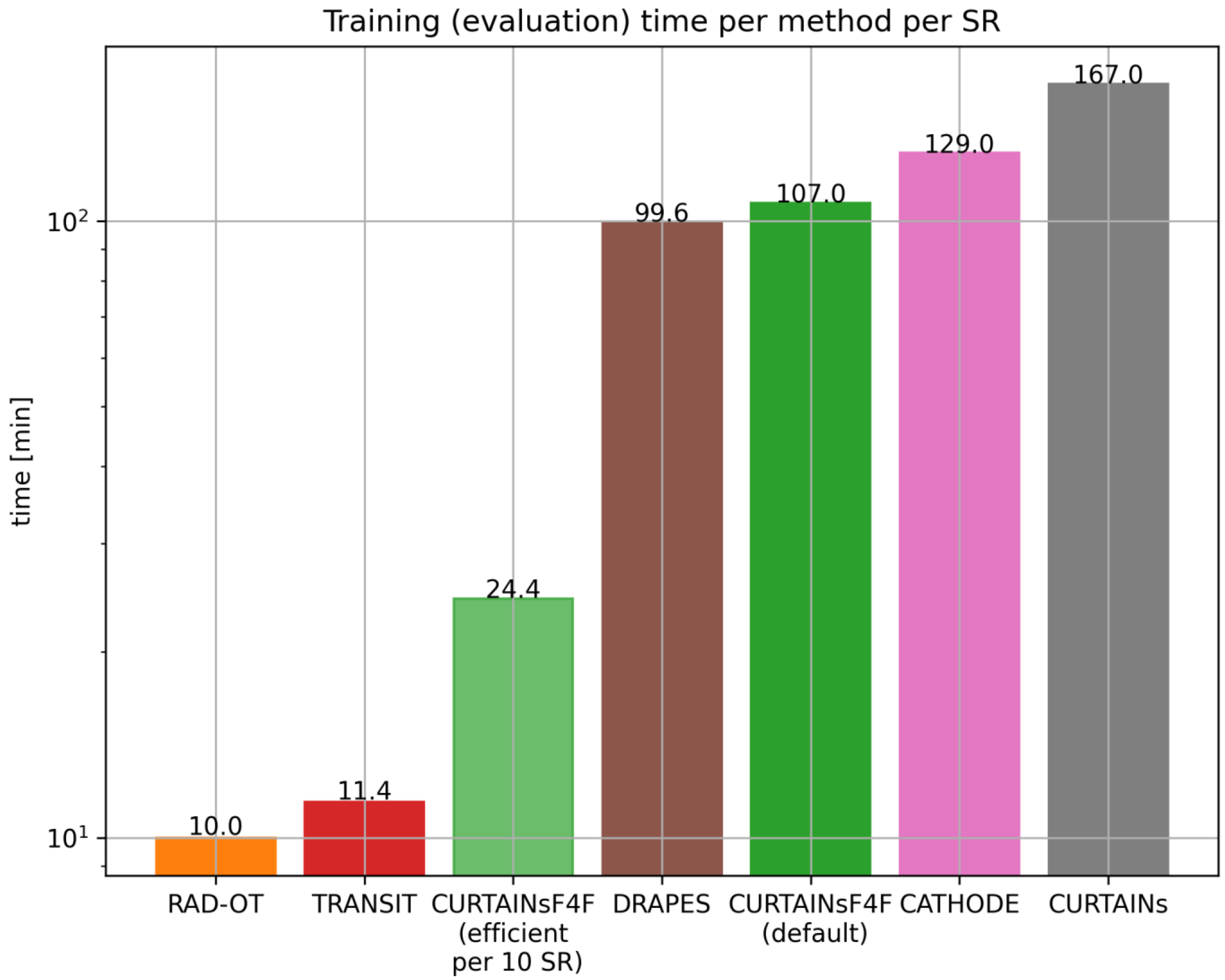




Speed comparison

- RAD-OT requires no training and is evaluated on CPU
- For most other methods generation time is negligible compared to training
- Trained using 1 NVIDIA® RTX 3080 GPU (+16 CPU cores)
- TRANSIT reaches 1 order of magnitude speedup compared to most other ML methods
- Efficient CURTAINsF4F relies on assumption that the base flow can be trained only once using all the data including SRs

*(All timing results for comparison come from ArXiv: 2305.04646 and ArXiv: 2407.19818)



Summary

- Template generation for weakly-supervised searches can be done without using flows or diffusion

➡ Just set a right objective!

Summary

- Template generation for weakly-supervised searches can be done without using flows or diffusion

➜ Just set a right objective!

- Transporting events instead of generating
  + appropriate architecture for this

➜ Efficient model!

Summary

- Template generation for weakly-supervised searches can be done without using flows or diffusion

➡ Just set a right objective!

- Transporting events instead of generating
  + appropriate architecture for this

➡ Efficient model!

- TRANSIT achieves performance competitive with SOTA (on LHCO) while taking only a fraction of computational resources

➡ Use for faster and better analysis!

Summary

- Template generation for weakly-supervised searches can be done without using flows or diffusion

  ➡ Just set a right objective!

- Transporting events instead of generating
  + appropriate architecture for this

  ➡ Efficient model!

- TRANSIT achieves performance competitive with SOTA (on LHCO) while taking only a fraction of computational resources

  ➡ Use for faster and better analysis!

- TRANSIT produces a set of mass decorrelated latent variables that mitigate background sculpting in CWoLa

  ➡ alternative to LaCATHODE!

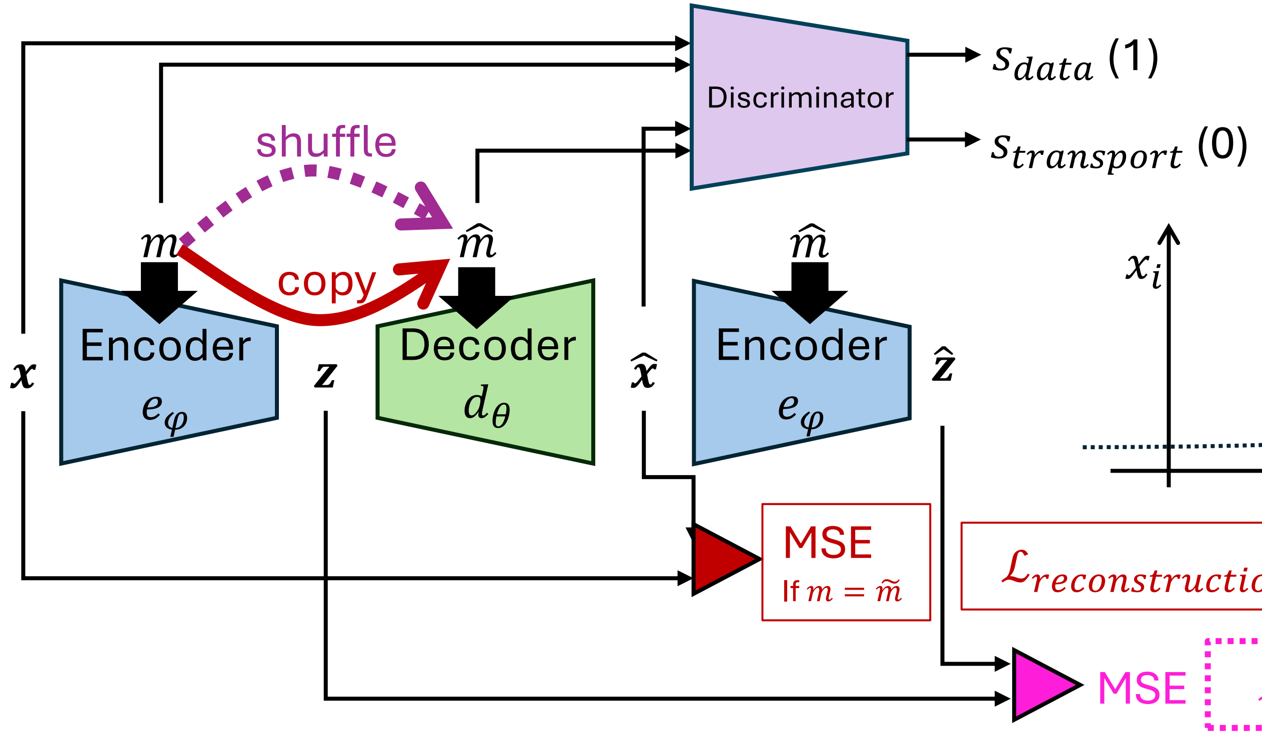
Outlook

- The method may be extended to any data format and number of variables using appropriate architecture, can use transformers to transport full particle clouds (WIP)
- Expect to get even higher speedups with code optimisation and graph compilation
- Looking forward to apply it in the next exotic search and SkyCURTAINS
- ArXive and code coming soon

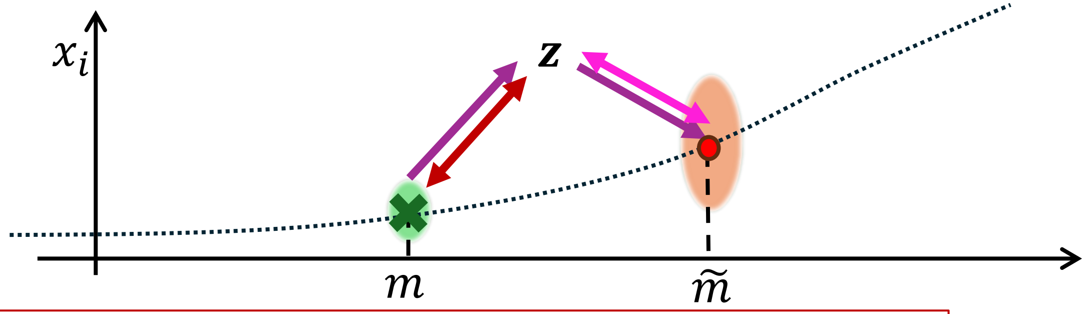

Thanks for your
attention!
DALL-E 3 interpretation

Backup

Structure and losses



$$\mathcal{L}_{transport} = -\mathcal{L}_{discriminator} = -BCE =$$
$$= 0.5\log(s_{data}) + 0.5\log(1 - s_{transport})$$
$$\Leftrightarrow \min(D_{KL}(p(\boldsymbol{x}|m)||\hat{p}_{\theta,\varphi}(\hat{\boldsymbol{x}}|\hat{m})))$$

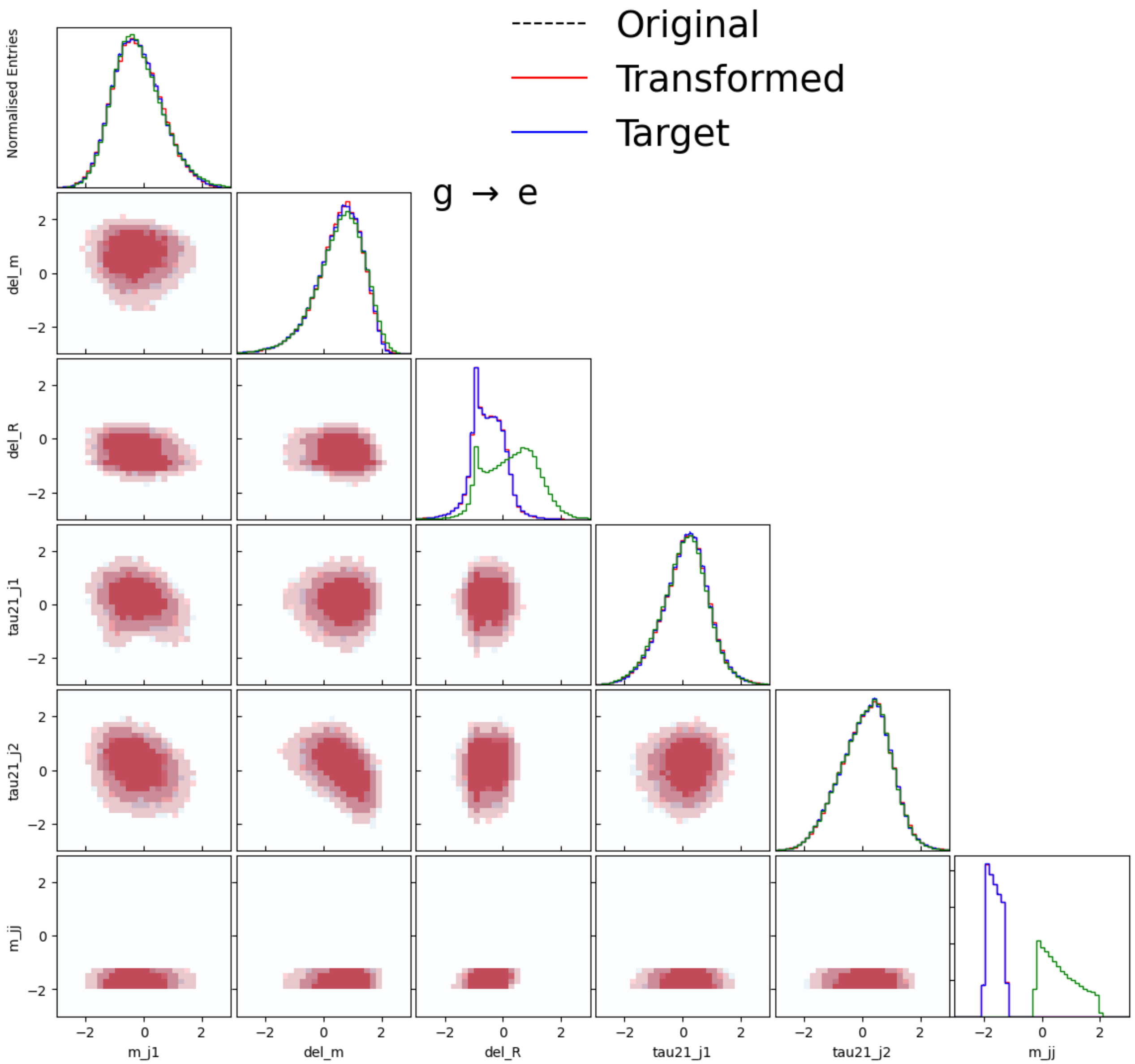


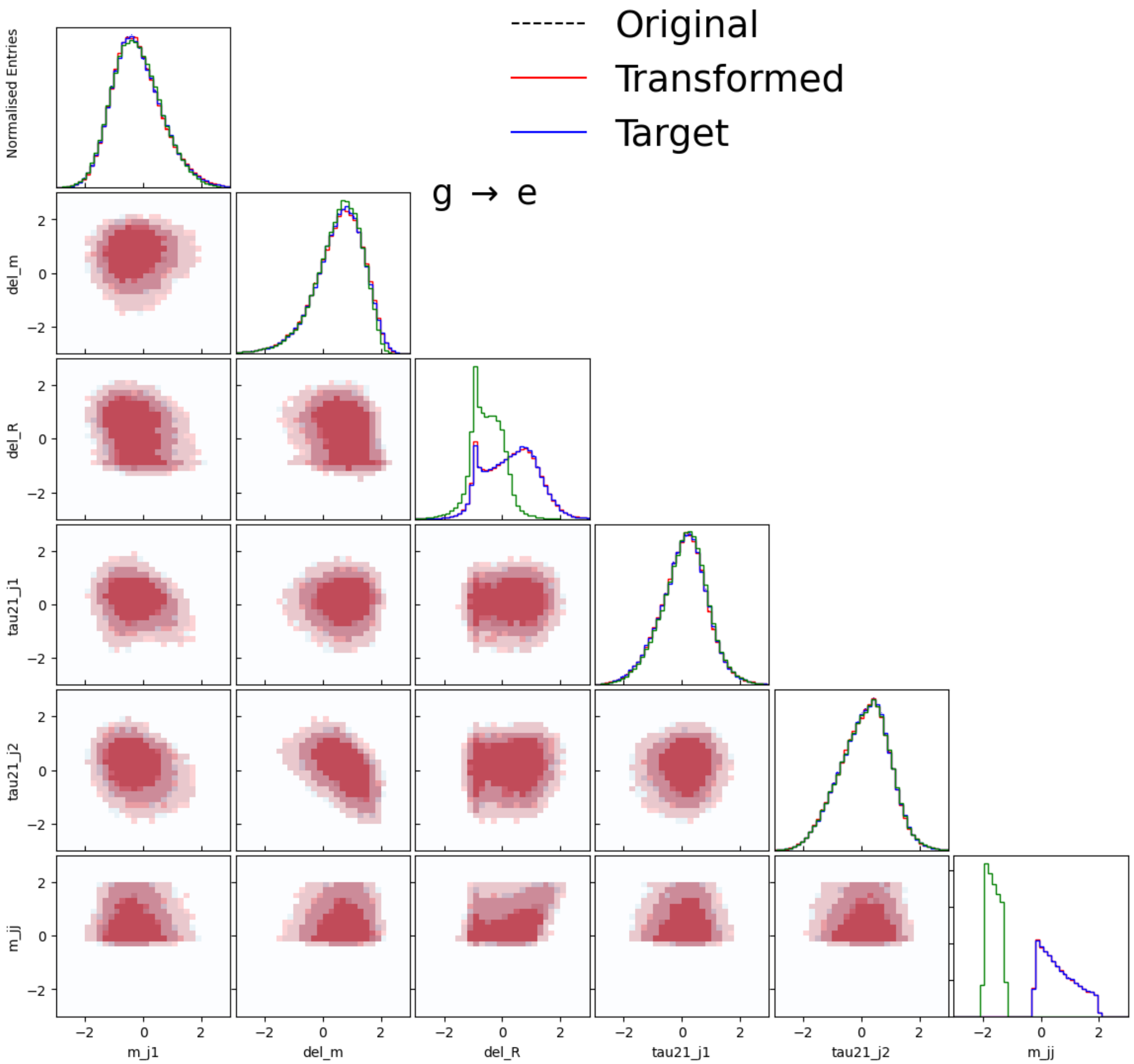
$$\mathcal{L}_{reconstruction} = \mathbb{E}_{p(x,m)}[\|d_\theta(e_\varphi(\boldsymbol{x}, m), m) - \boldsymbol{x}\|]$$

$$\mathcal{L}_{consistency} = \mathbb{E}_{p(z)p(m)}[\|e_\varphi(d_\theta(\boldsymbol{z}, m), m) - \boldsymbol{z}\|]$$

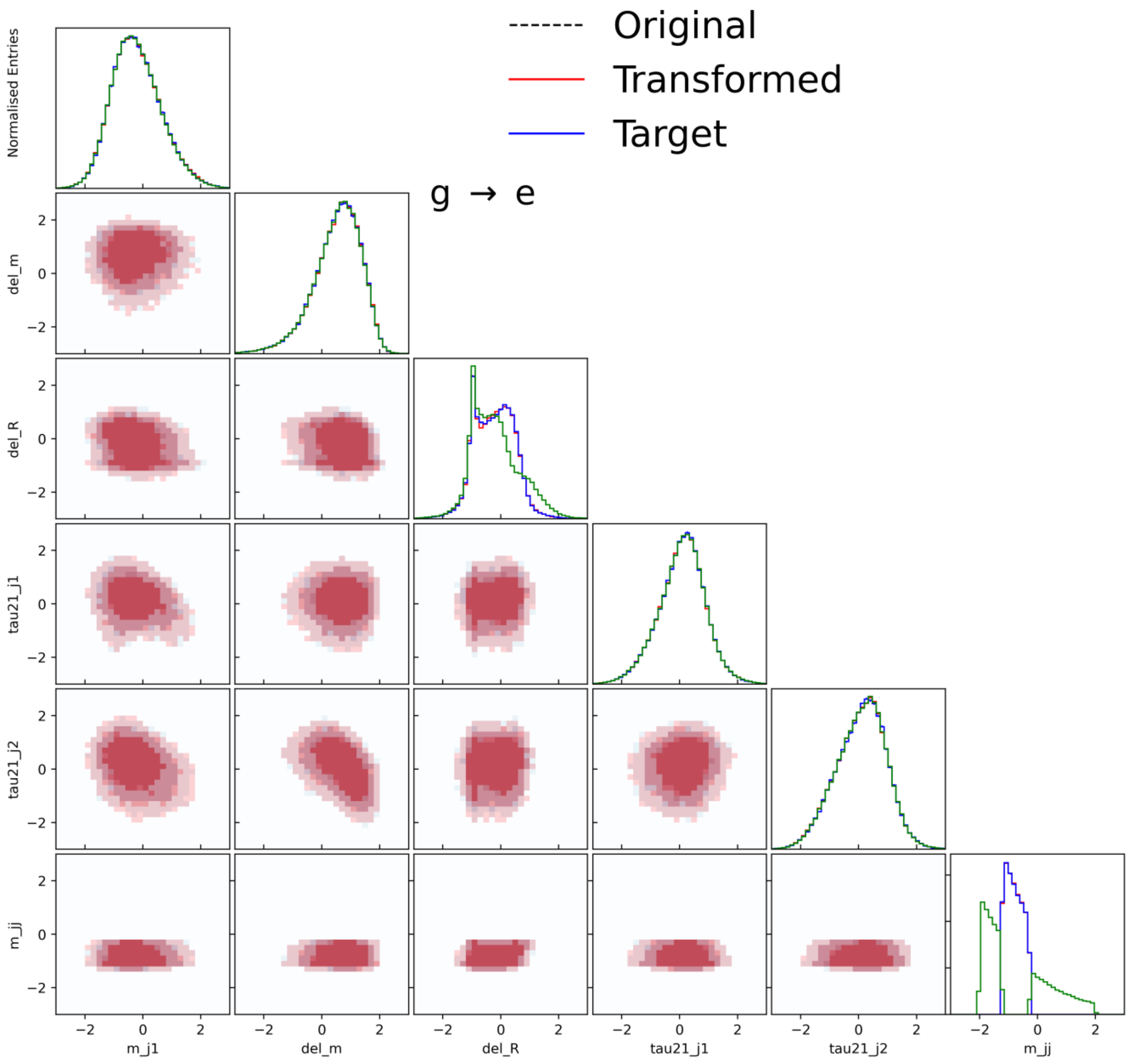
- Propagate through network twice once with $\tilde{m} = m$ once with $\tilde{m} = \text{shuffle}(m)$
- Encoder+Decoder use the negative of Discriminator loss
- Update Discriminator 1 step per 1 step of Encoder+Decoder
- If $\mathcal{L}_{discriminator}$ > ln(2) (random classifier) update only Discriminator!

Siband to sideband transport





Sideband to SR transport



Mass sculpting: Background only

Select only background using CWoLa score threshold.

Latent features have practically no correlation