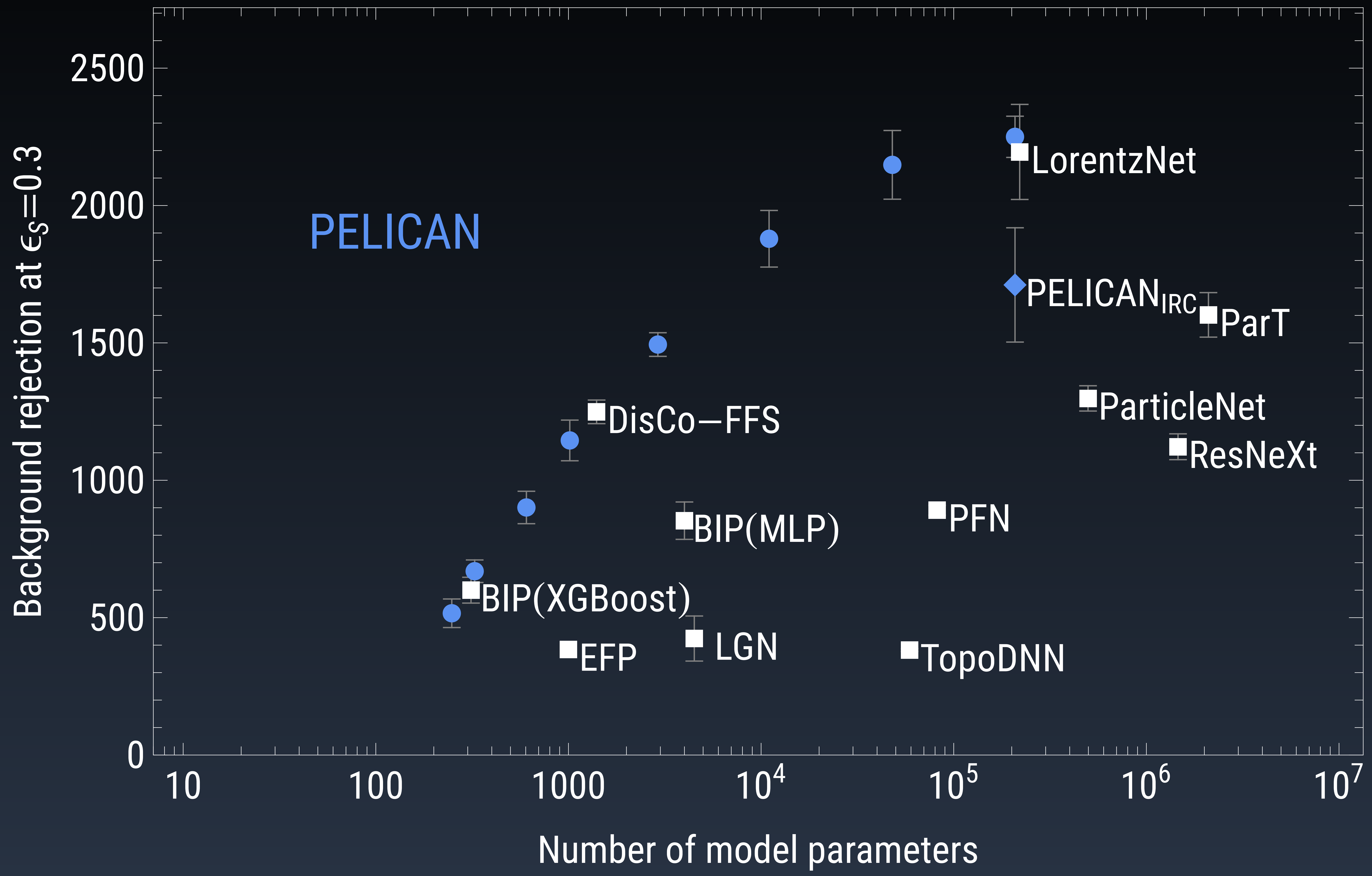
Achieving linear scaling: CARDINAL

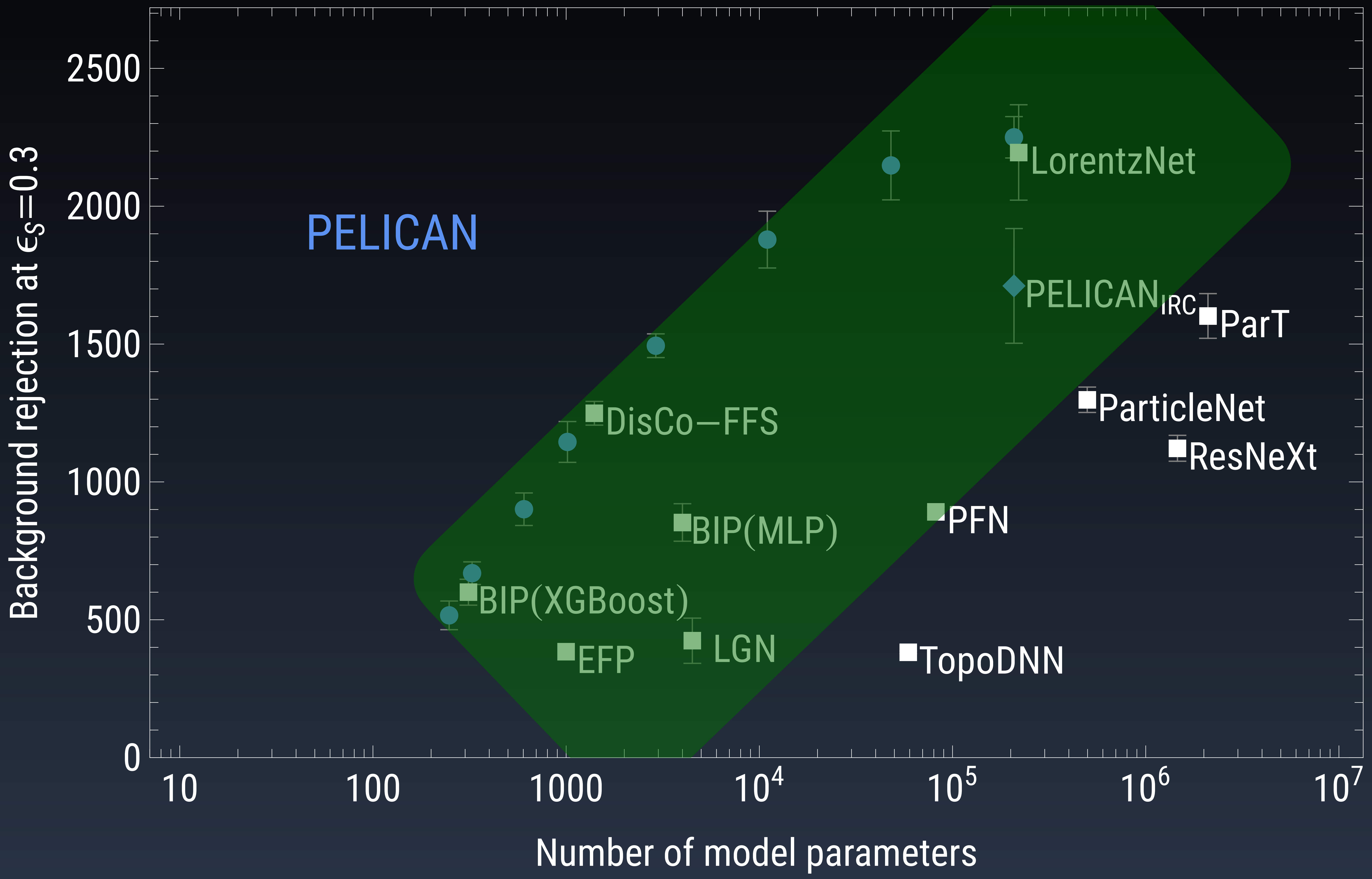
Alexander Bogatskiy
DSI, UChicago

7/30/2025

ML top taggers

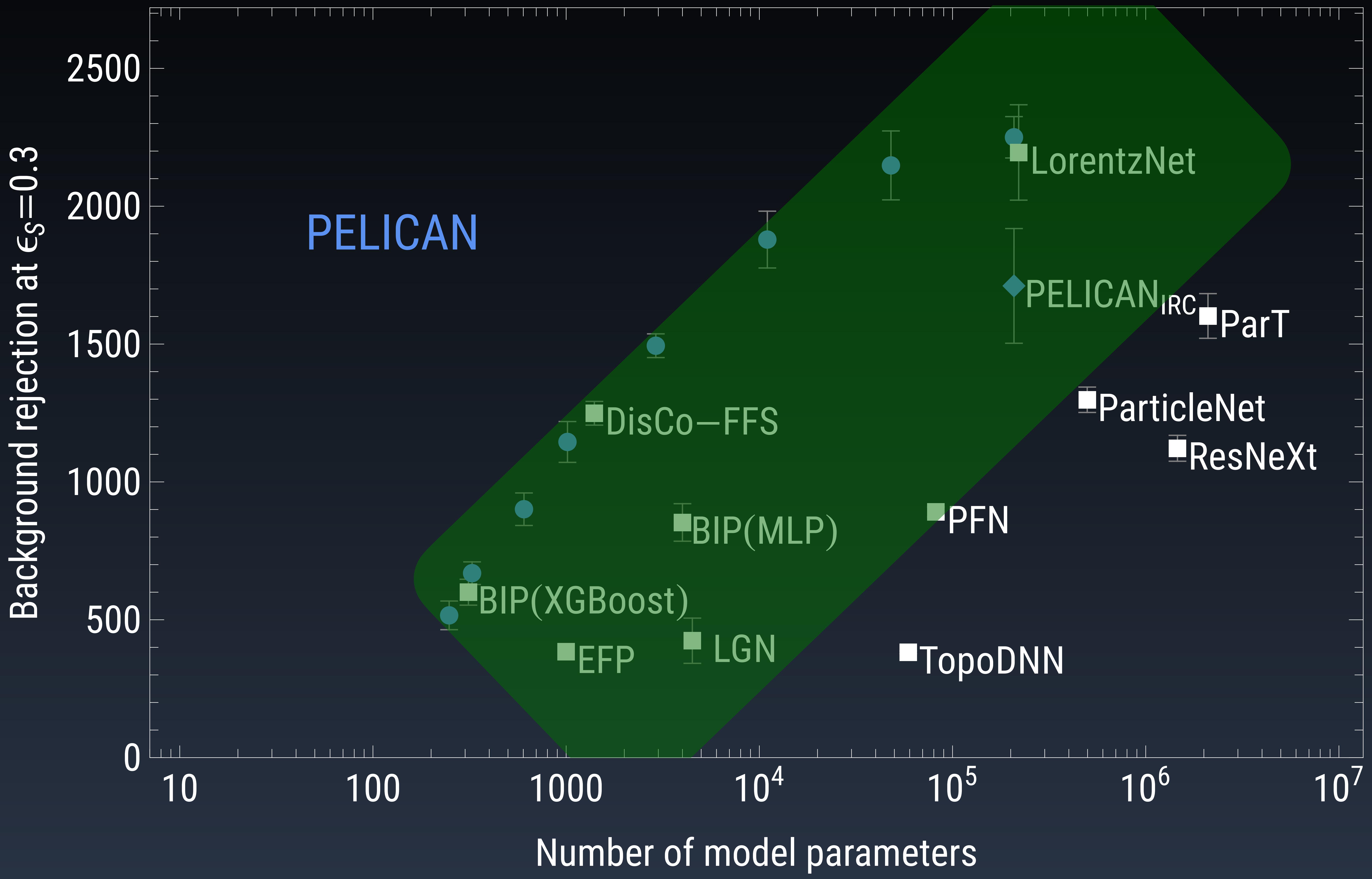


ML top taggers



ML top taggers

Lorentz Invariance



Scaling behavior

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$

$$\mathcal{O}(N^2)$$

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$
- **ParT** introduced large transformer models

$$\Big] \; \mathcal{O}(N^2)$$

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$
- **ParT** introduced large transformer models
- **ParticleNet** is a k-NN dynamic GNN

$\mathcal{O}(N^2)$

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$
- **ParT** introduced large transformer models
- **ParticleNet** is a k-NN dynamic GNN

$\mathcal{O}(N^2)$

$\mathcal{O}(N^2)$

worst case

Scaling behavior

- **LorentzNet** introduced pairwise features $m_{ij} = p_i \cdot p_j$
- PELICAN succeeded using *only* $p_i \cdot p_j$

] $\mathcal{O}(N^2)$

- **ParT** introduced large transformer models
- **ParticleNet** is a k-NN dynamic GNN

] $\mathcal{O}(N^2)$

worst case

Reason: permutation/Lorentz equivariance

The trade-off

Scalability ⟷ **Symmetry**

The trade-off

Scalability ⟷ **Symmetry**

Is there a workaround?

1. Lorentz invariance

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants
  - Open problem (my guess is it has no satisfying solution)

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants
    - Open problem (my guess is it has no satisfying solution)
- **Idea 2:** **canonicalize** 4-momenta

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants
    - Open problem (my guess is it has no satisfying solution)
- **Idea 2:** **canonicalize** 4-momenta
    - Suffers from singularities, but in jet physics it is possible to place the singularities in unphysical regions

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants
  - Open problem (my guess is it has no satisfying solution)
- **Idea 2:** **canonicalize** 4-momenta
  - Suffers from singularities, but in jet physics it is possible to place the singularities in unphysical regions
  - Every event contains two "beams", say $e_\pm = (1,0,0,\pm 1)$. Canonicalize by BOOSTing to the *unique rest frame* of the event in which $e_+^y = e_+^z = e_-^z = 0$.

1. Lorentz invariance

- Pairwise features inherently are $\mathcal{O}(N^2)$ in number
- **Idea 1:** find a complete set of $\mathcal{O}(N)$ Lorentz invariants
  - Open problem (my guess is it has no satisfying solution)
- **Idea 2:** **canonicalize** 4-momenta
  - Suffers from singularities, but in jet physics it is possible to place the singularities in unphysical regions
  - Every event contains two "beams", say $e_\pm = (1,0,0,\pm 1)$. Canonicalize by BOOSTing to the *unique rest frame* of the event in which $e_+^y = e_+^z = e_-^z = 0$.
  - Possible as long as the total momentum is timelike and not parallel to the beam axis.

1. Lorentz invariance

$$e_+, e_-, p_1, \ldots, p_N \quad \mapsto \quad \bar{e}_+, \bar{e}_-, \bar{p}_1, \ldots, \bar{p}_N$$

1. Lorentz invariance

$$e_+, e_-, p_1, \ldots, p_N \quad \mapsto \quad \bar{e}_+, \bar{e}_-, \bar{p}_1, \ldots, \bar{p}_N$$

Dimensionality unchanged

1. Lorentz invariance

$$e_+, e_-, p_1, \ldots, p_N \quad \mapsto \quad \bar{e}_+, \bar{e}_-, \bar{p}_1, \ldots, \bar{p}_N$$

Dimensionality unchanged

Pairwise constituent-constituent features replaced by constituent-jet features

2. Permutation invariance

2. Permutation invariance

Universal approximators — Deep Sets — are rarely used.

2. Permutation invariance

Universal approximators — Deep Sets — are rarely used.

Common solutions are *equivariant*

2. Permutation invariance

Universal approximators — Deep Sets — are rarely used.

Common solutions are *equivariant*

- Avoids “expressivity bottlenecks” such as in Deep Sets

2. Permutation invariance

Universal approximators — Deep Sets — are rarely used.

Common solutions are *equivariant*

- Avoids “expressivity bottlenecks” such as in Deep Sets
- This contributes to their complexity and *limits the expressivity*

2. Permutation invariance

Universal approximators — Deep Sets — are rarely used.

Common solutions are *equivariant*

- Avoids "expressivity bottlenecks" such as in Deep Sets
- This contributes to their complexity and *limits the expressivity*
- E.g. MPNNs, GNNs, Transformers are "1-GNN's"

2. Permutation invariance

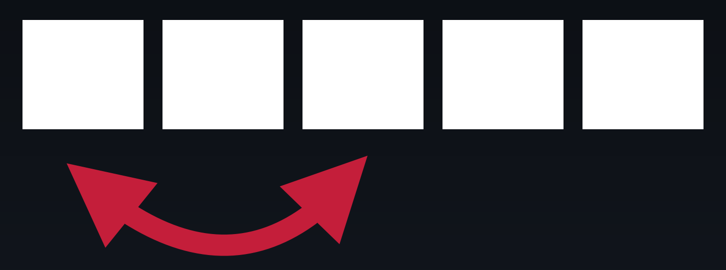
Universal approximators — Deep Sets — are rarely used.

Common solutions are *equivariant*

- Avoids “expressivity bottlenecks” such as in Deep Sets
- This contributes to their complexity and *limits the expressivity*
- E.g. MPNNs, GNNs, Transformers are “1-GNN’s”
- **PELICAN** is a “2-GNN”, hence more expressive

2. Permutation invariance

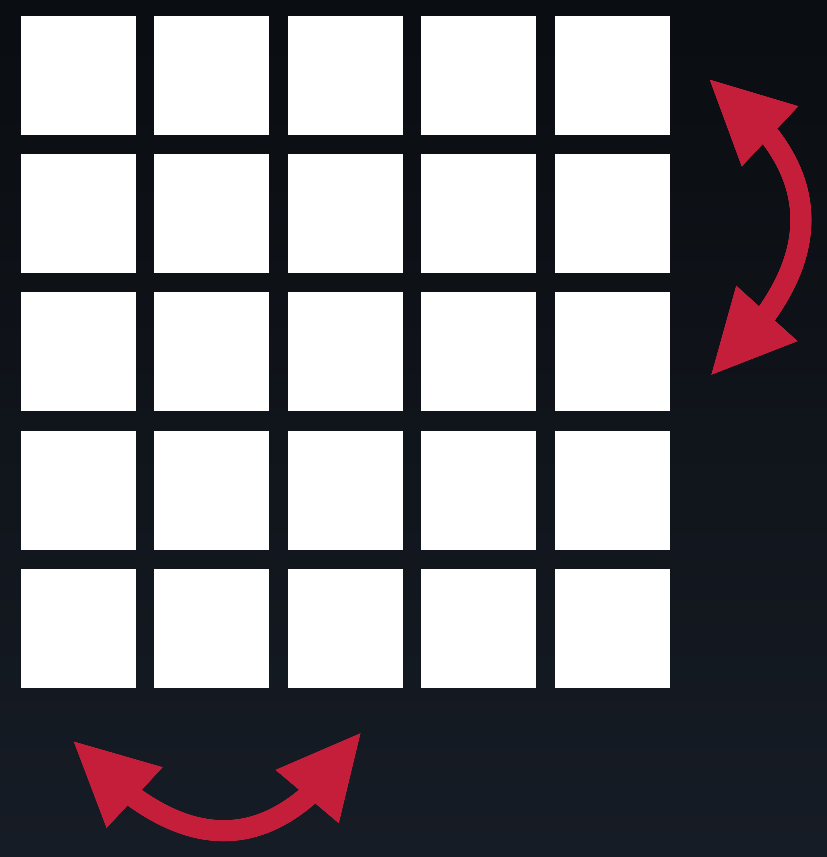
1-GNN



2. Permutation invariance

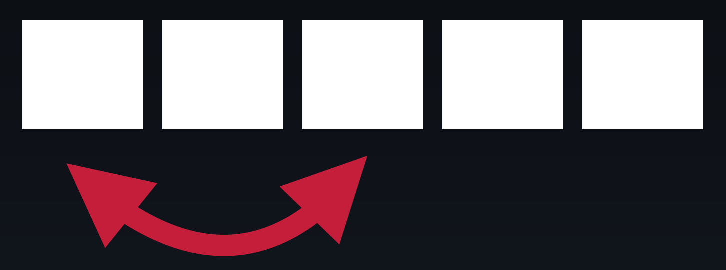
1-GNN

2-GNN

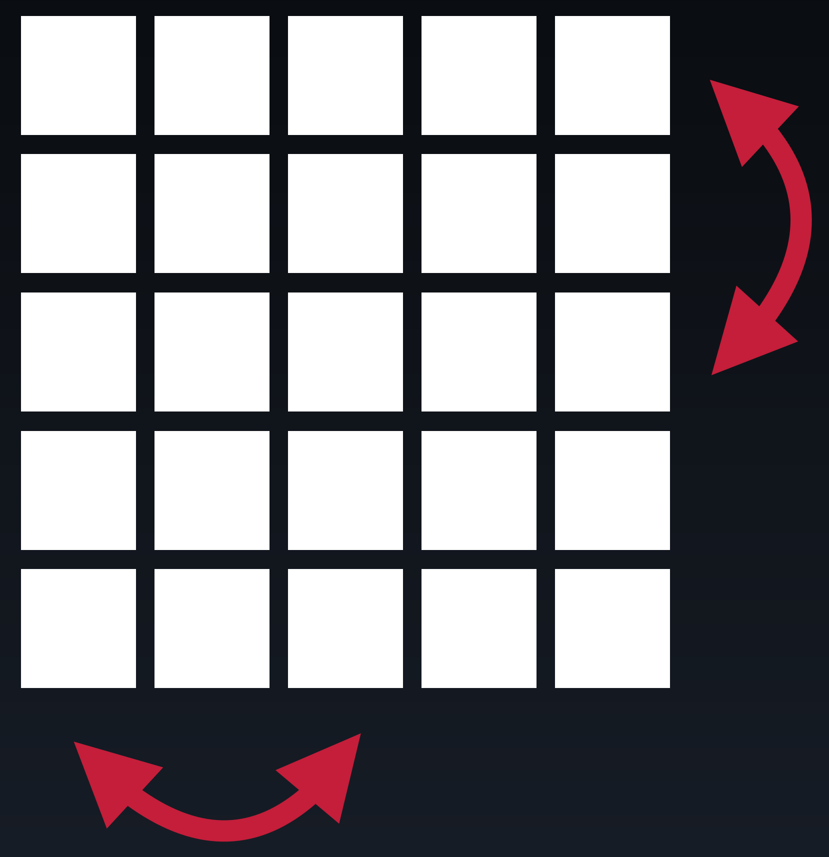


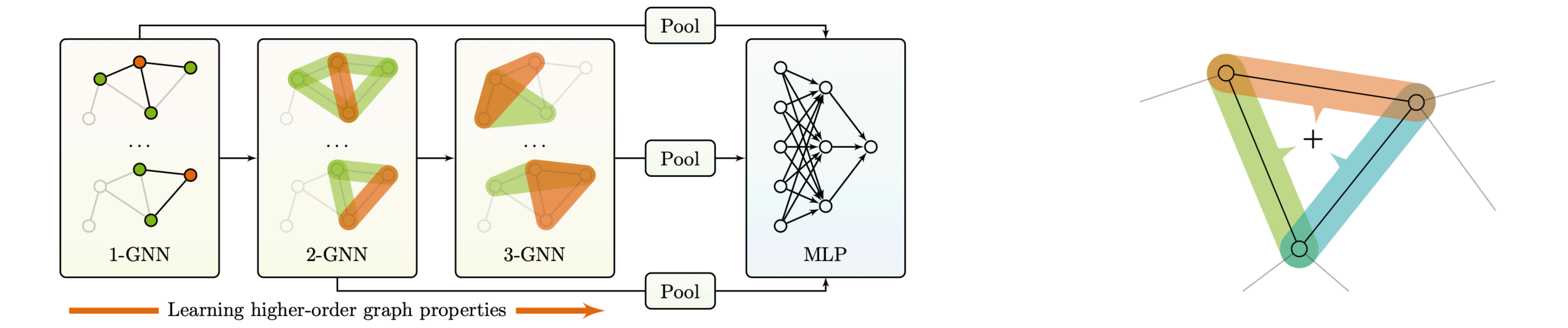
2. Permutation invariance

1-GNN



2-GNN





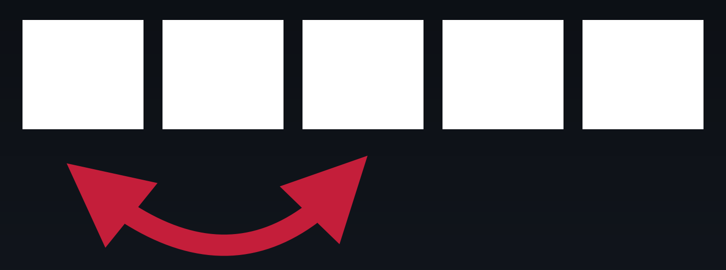
(a) Hierarchical 1-2-3-GNN network architecture

(b) Pooling from 2- to 3-GNN.

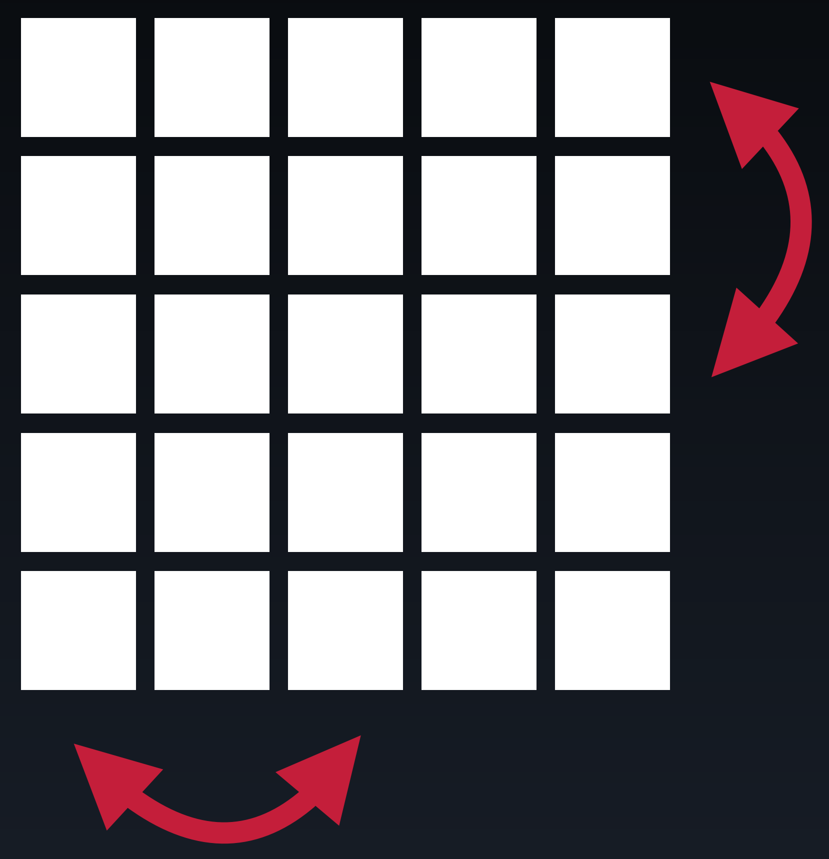
[arXiv:1810.02244]

2. Permutation invariance

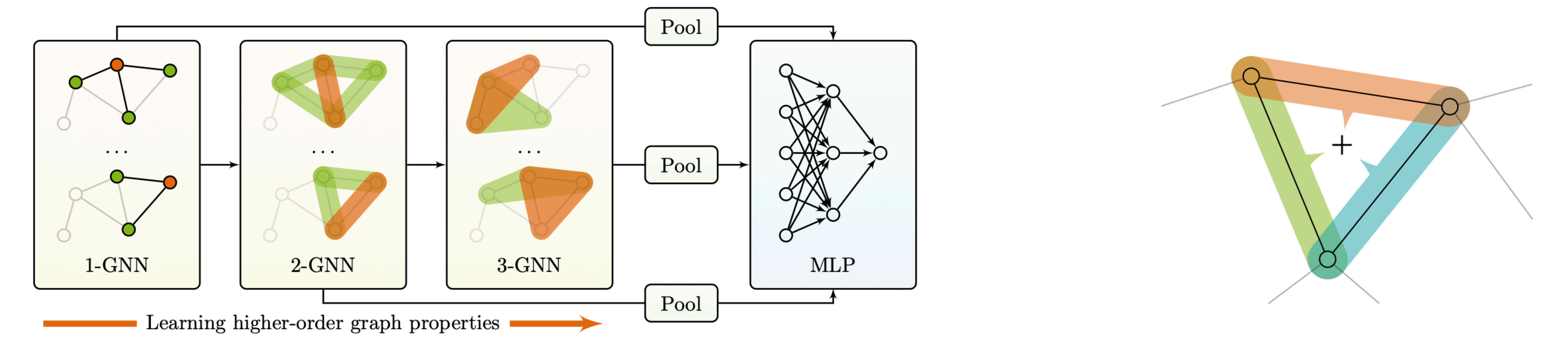
1-GNN



2-GNN



We pay for equivariance and expressivity in complexity



(a) Hierarchical 1-2-3-GNN network architecture

(b) Pooling from 2- to 3-GNN.

[arXiv:1810.02244]

2. Permutation invariance

2. Permutation invariance

- **Idea:** use a permutation-invariant embedder

2. Permutation invariance

- **Idea:** use a permutation-invariant embedder
  - e.g. dot every constituent's feature vector x_i into M fixed random vectors v_j and sort over the constituents — a very cheap and piecewise linear embedding

$$\mathrm{Sort}_i\{x_i \cdot v_j\}$$

CARDINAL

CARDINAL

Two variants:

CARDINAL

Two variants:

$$\{p_i\} \mapsto \{\bar{p}_i\} \longrightarrow \boxed{\textbf{Inv. Emb.}} \longrightarrow \boxed{\textbf{MLP}}$$

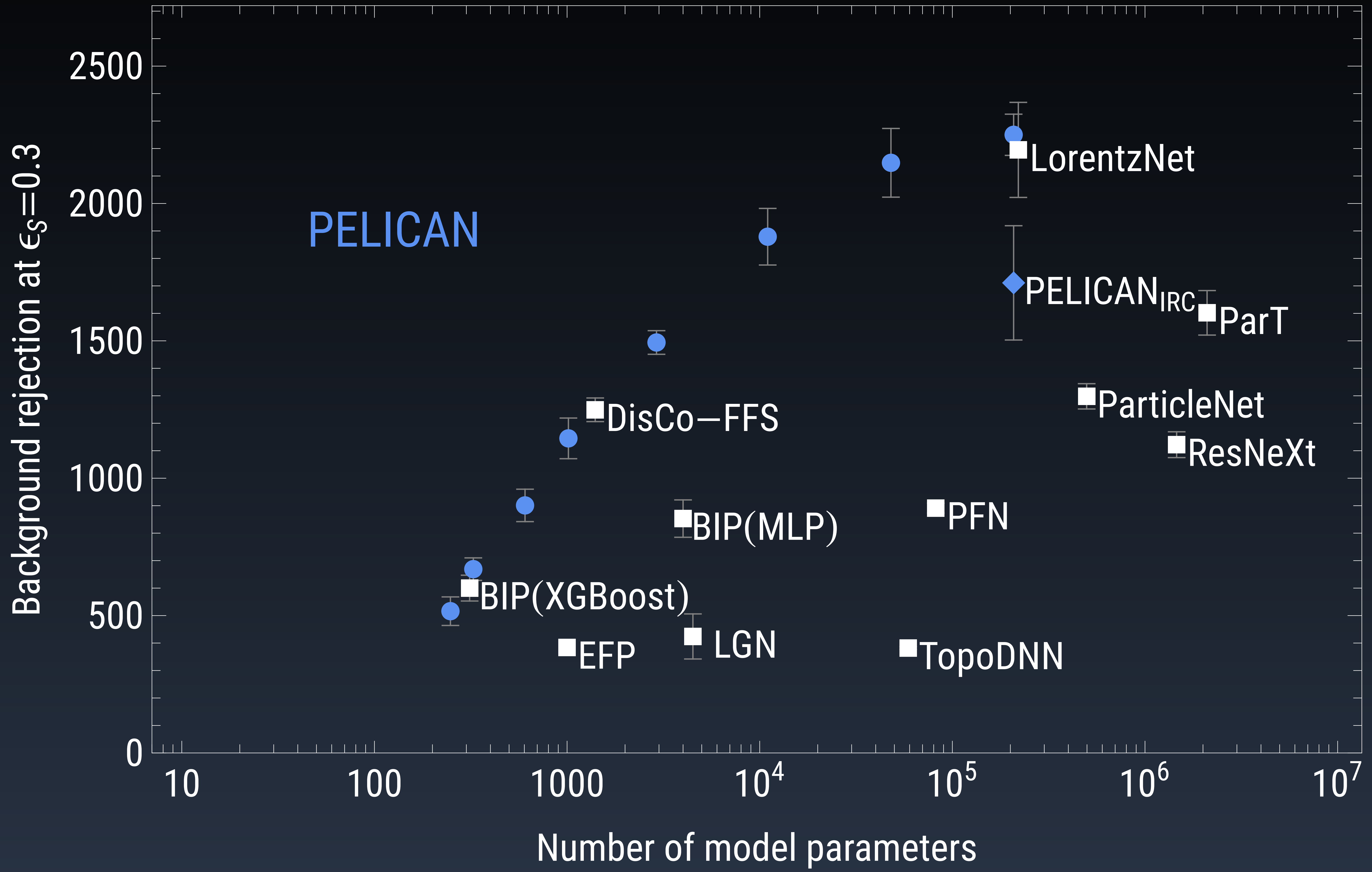
CARDINAL

Two variants:

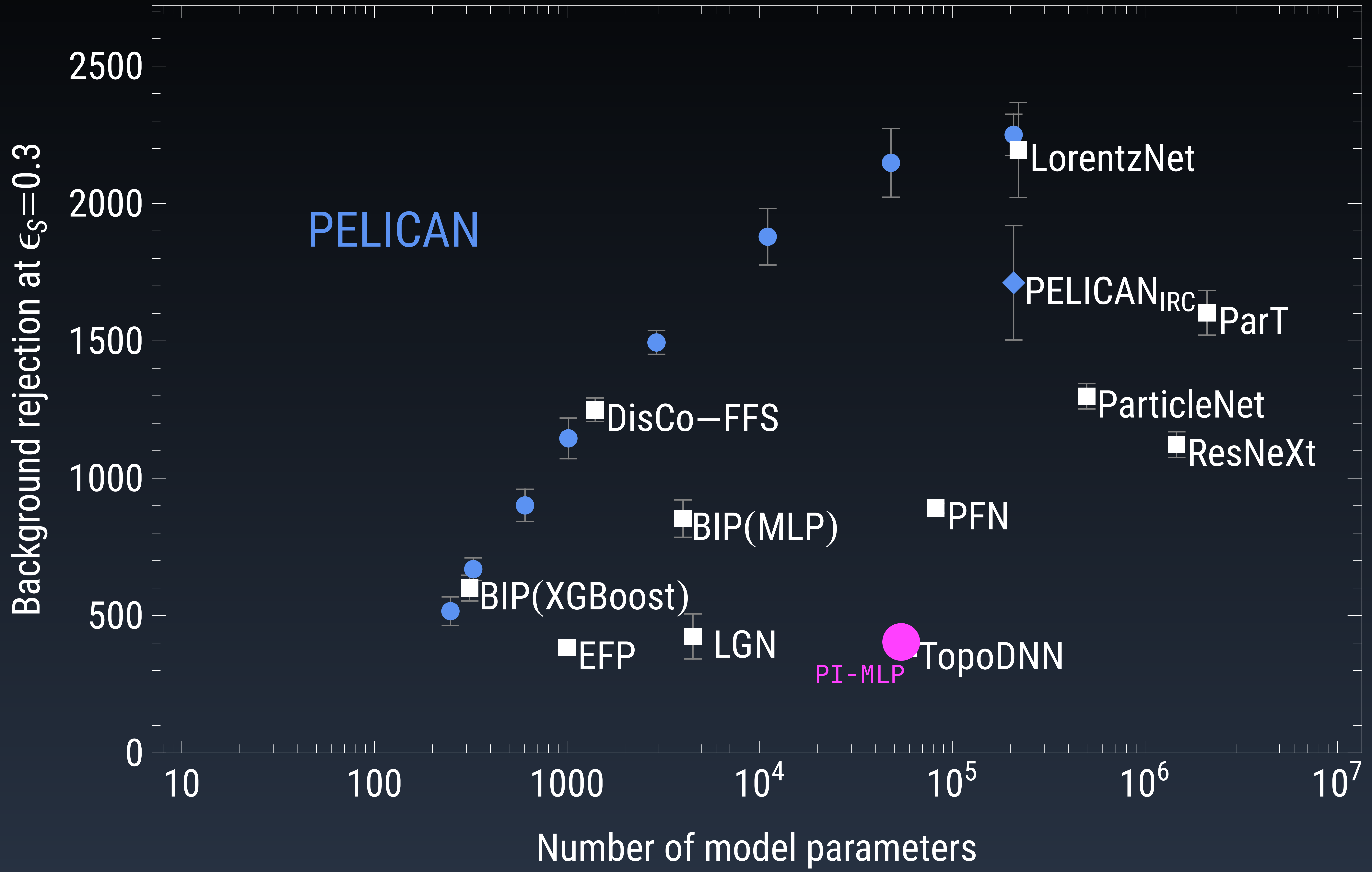
$$\{p_i\} \mapsto \{\bar{p}_i\} \longrightarrow \boxed{\textbf{Inv. Emb.}} \longrightarrow \boxed{\textbf{MLP}}$$

$$\{p_i\} \mapsto \{\bar{p}_i\} \longrightarrow \boxed{\textbf{MPNN}}$$

Top tagging quality



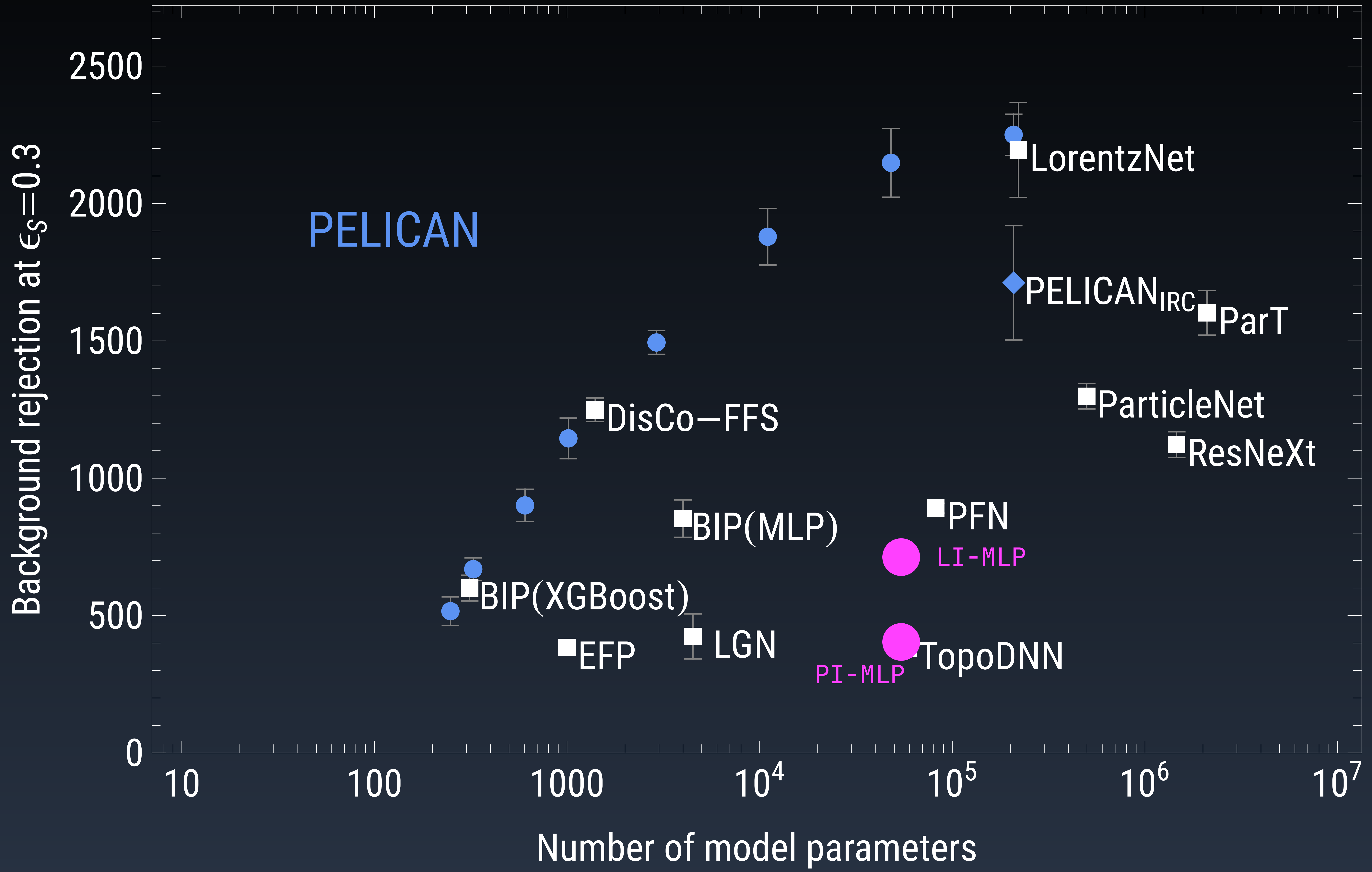
Top tagging quality



PI-MLP

$\{p_i\} \to$ Inv. Emb. $\to$ MLP

Top tagging quality



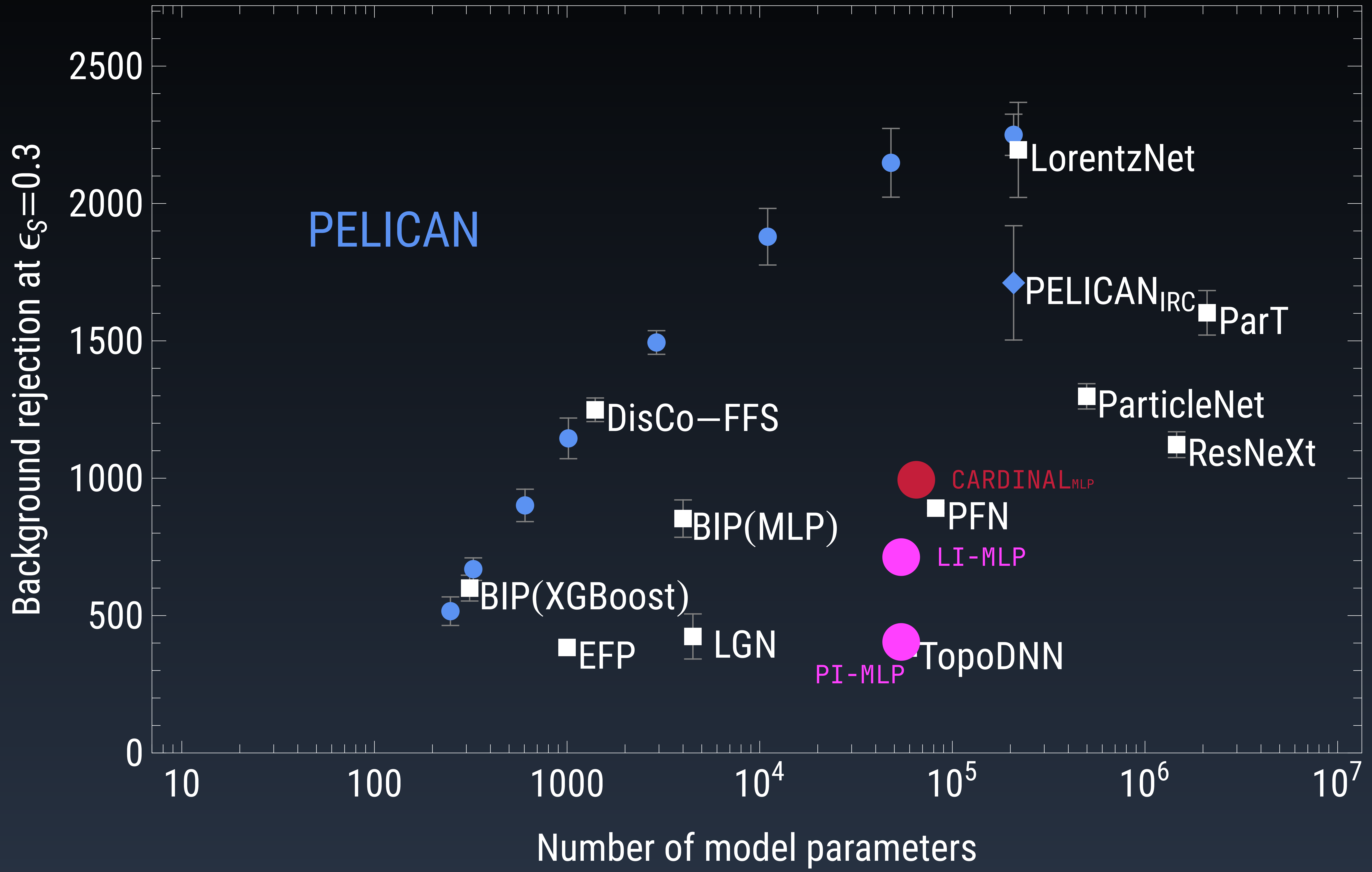
PI-MLP

$\{p_i\} \to$ Inv. Emb. $\to$ MLP

LI-MLP

$\{\bar{p}_i\} \to$ MLP

Top tagging quality



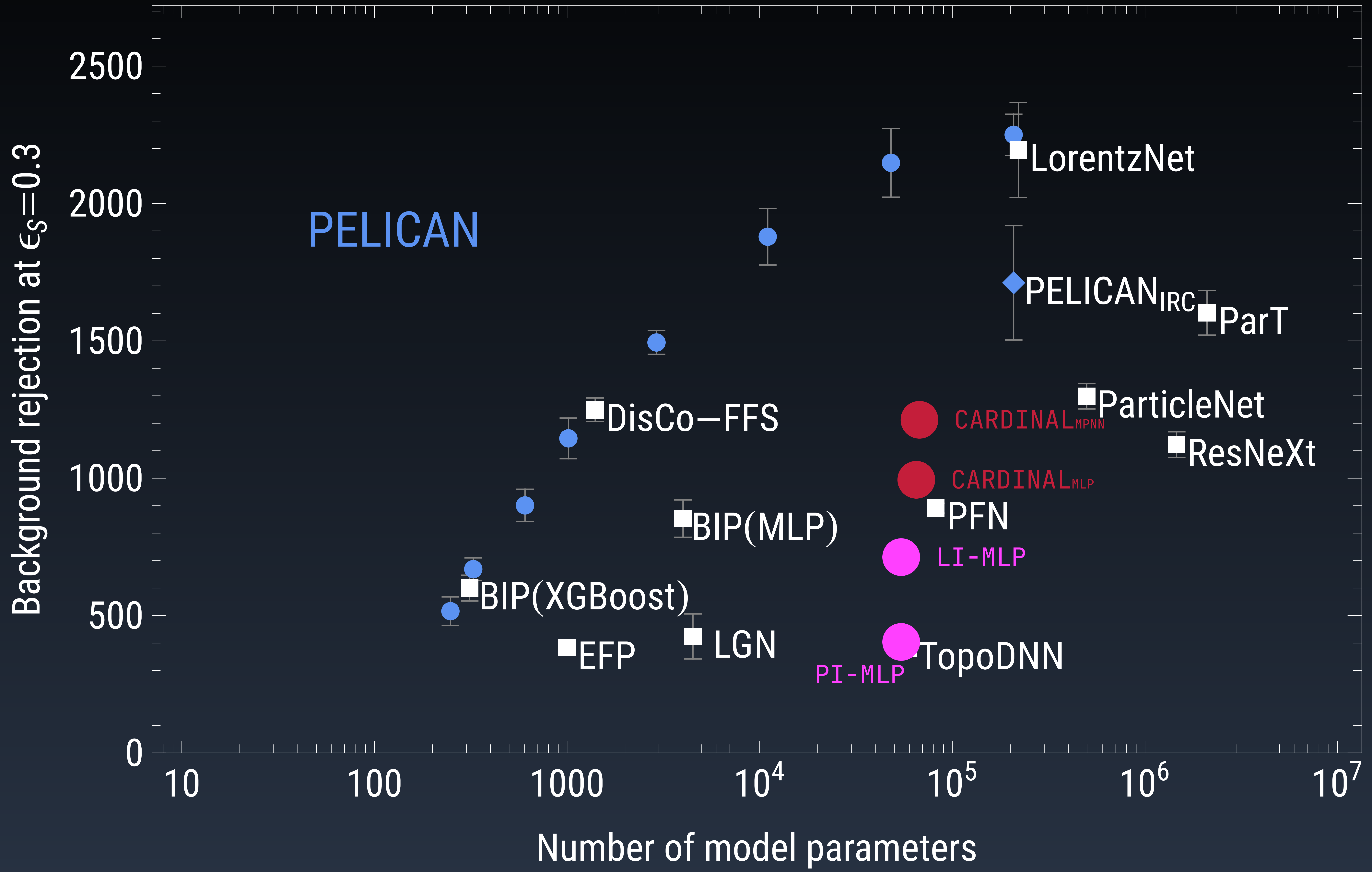
PI-MLP

$\{p_i\} \to$ Inv. Emb. $\to$ MLP

LI-MLP

$\{\bar{p}_i\} \to$ MLP

Top tagging quality



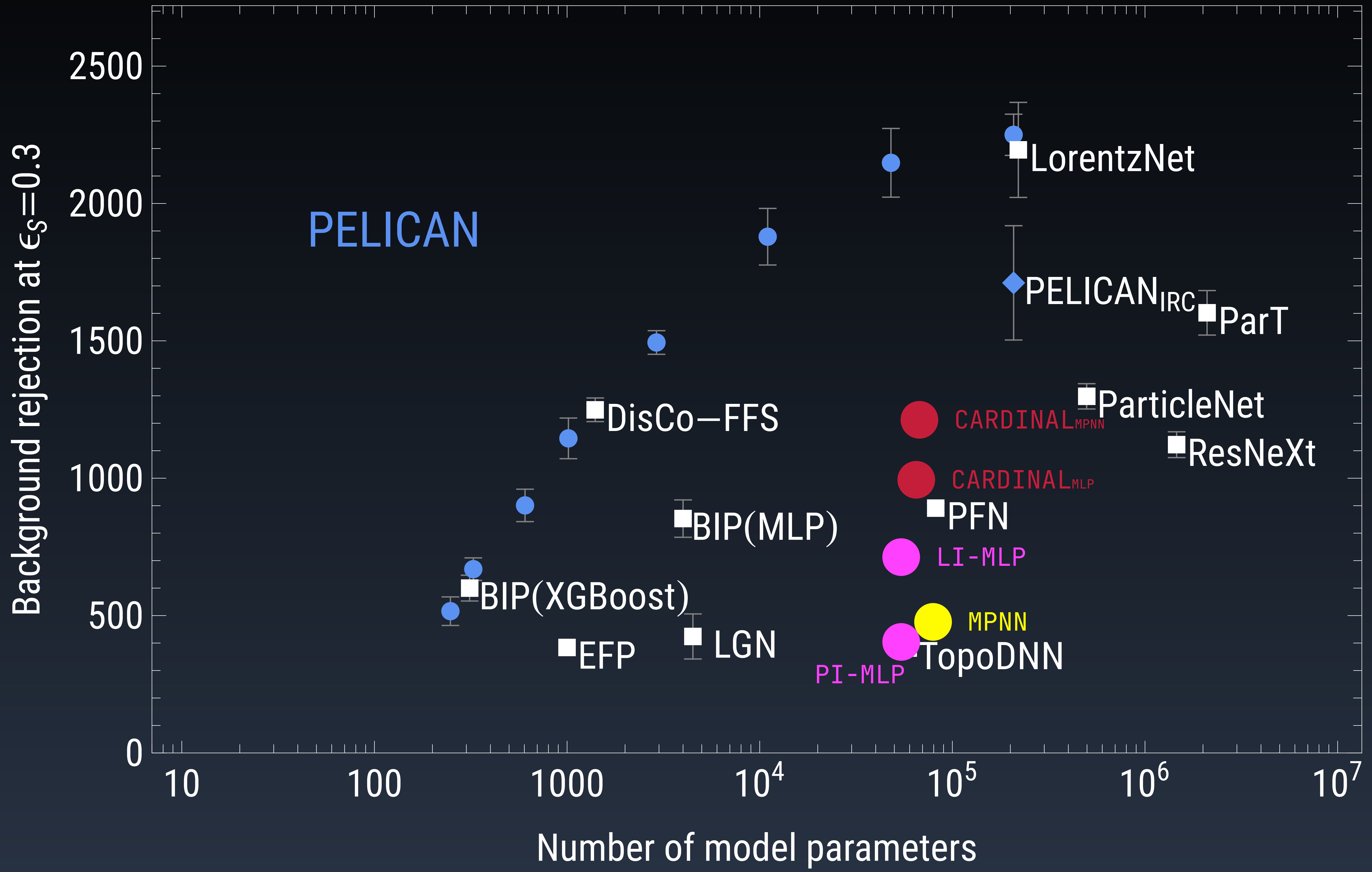
PI-MLP

$\{p_i\} \rightarrow$ Inv. Emb. $\rightarrow$ MLP

LI-MLP

$\{\bar{p}_i\} \rightarrow$ MLP

Top tagging quality



PI-MLP

$\{p_i\} \to$ Inv. Emb. $\to$ MLP

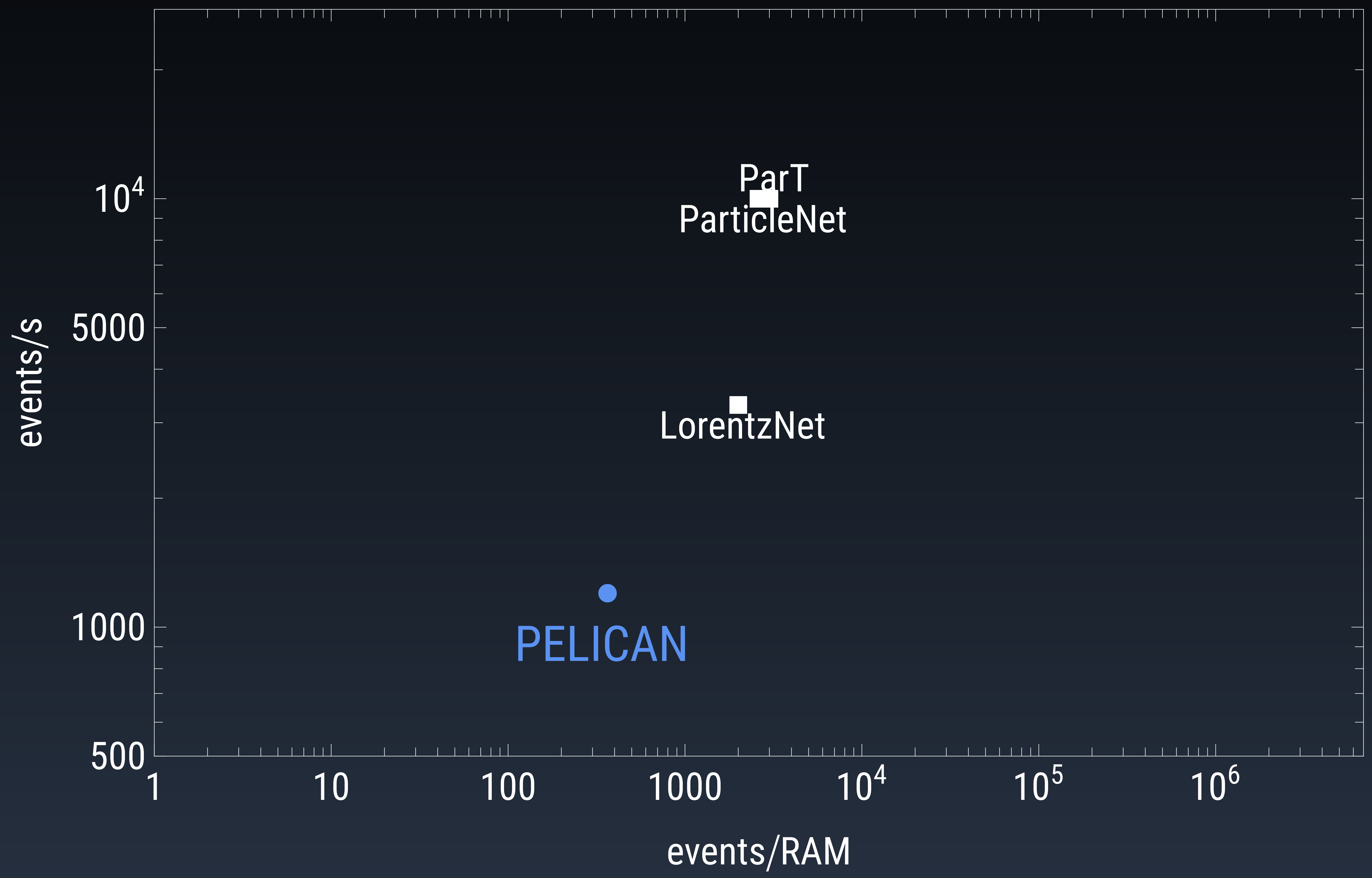
LI-MLP

$\{\bar{p}_i\} \to$ MLP

MPNN

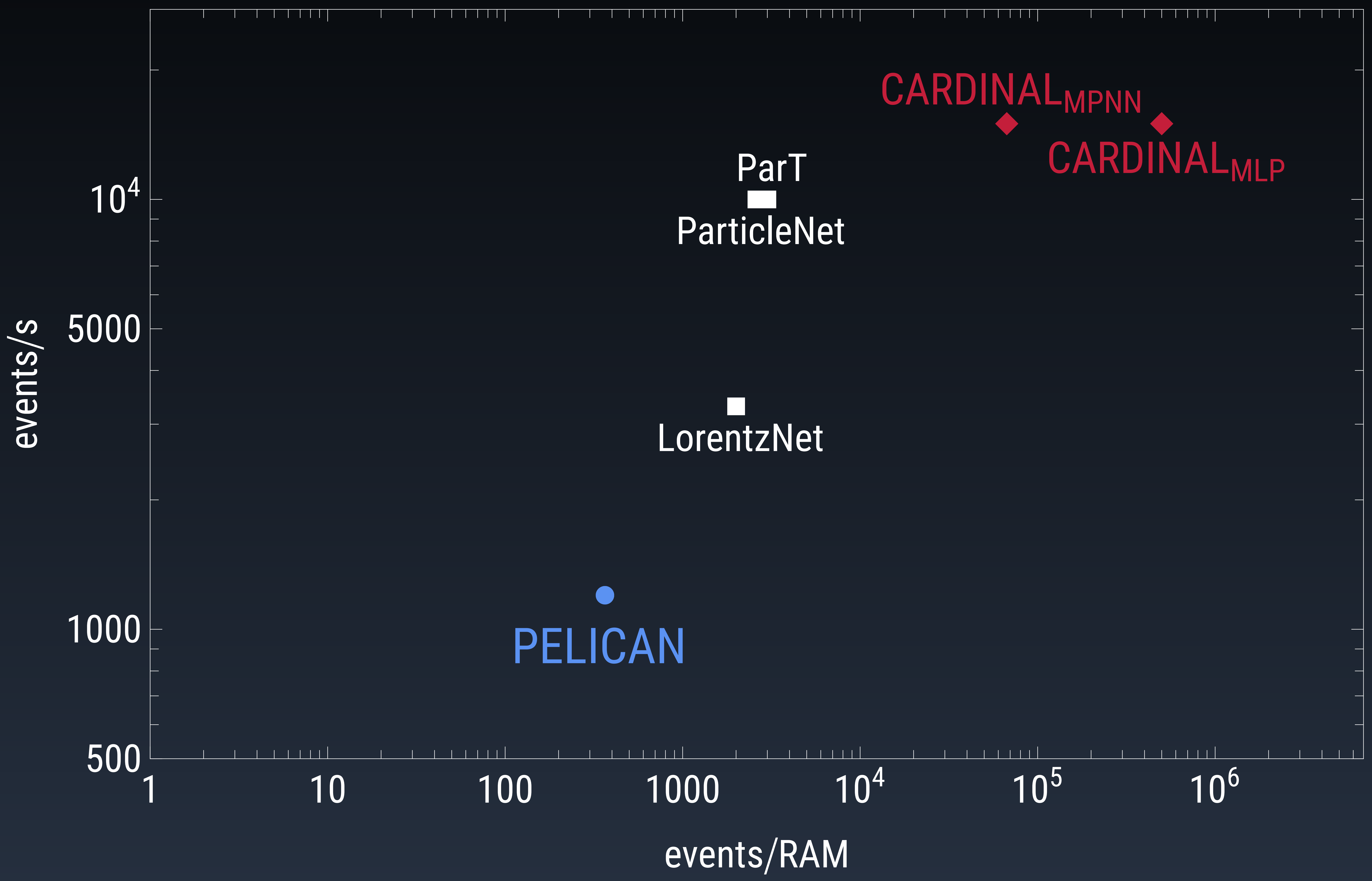
$\{p_i\} \to$ MPNN

Resource efficiency



(A100 80GB)

Resource efficiency



(A100 80GB)

Applications

Applications

- Pile-up

Applications

- Pile-up
- Full event reconstruction

Applications

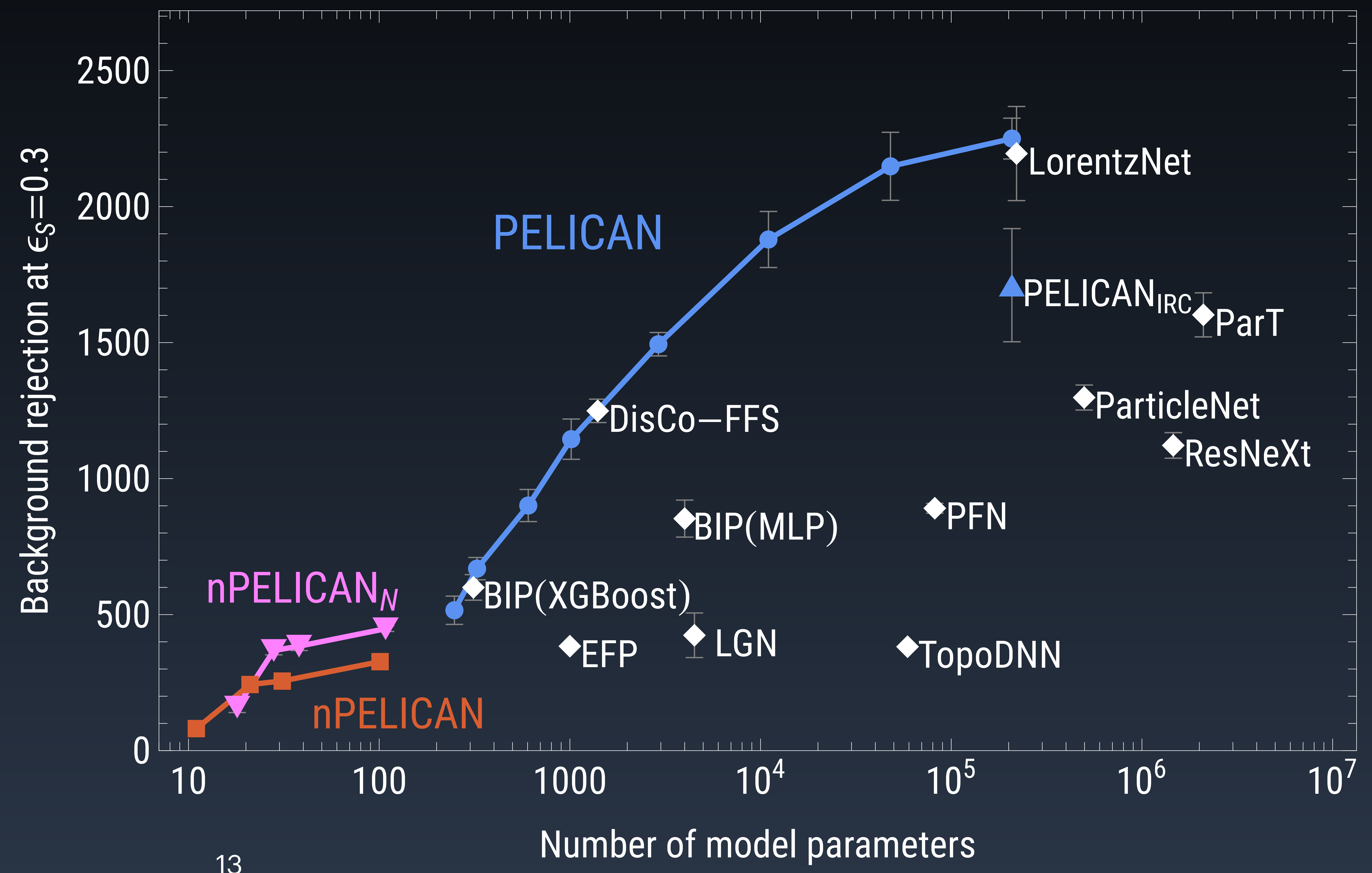
- Pile-up
- Full event reconstruction
- Heavy ion collisions

Applications

- Pile-up
- Full event reconstruction
- Heavy ion collisions
- Small networks for FPGAs

Applications

- Pile-up
- Full event reconstruction
- Heavy ion collisions
- Small networks for FPGAs



Discussion

Discussion

- Why do pairwise feature inputs help?

Discussion

- Why do pairwise feature inputs help?
- Why does Lorentz invariance (vs none) help?

Discussion

- Why do pairwise feature inputs help?
- Why does Lorentz invariance (vs none) help?
- Why does permutation equivariance (vs invariance) help?

Discussion

- Why do pairwise feature inputs help?
- Why does Lorentz invariance (vs none) help?
- Why does permutation equivariance (vs invariance) help?
- The "optimality" of transformers is a myth