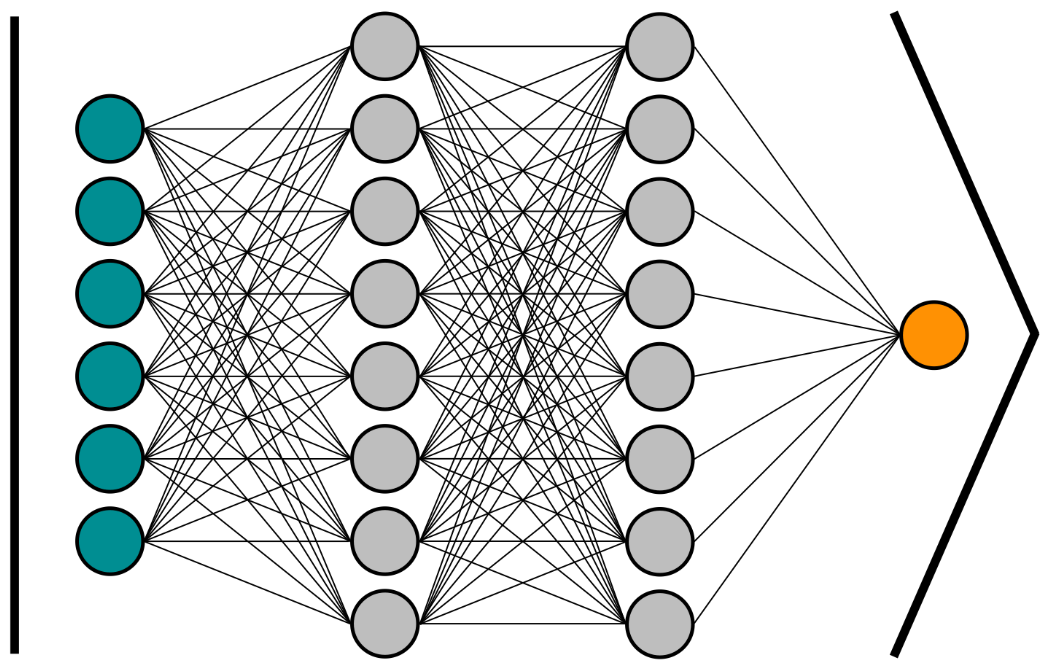


Ultracold Fermi gases with Neural-Network Quantum States

Jane Kim
Ohio University (MSU/FRIB)
ECT* Many-Body Quantum Physics with Machine Learning
5 September 2023

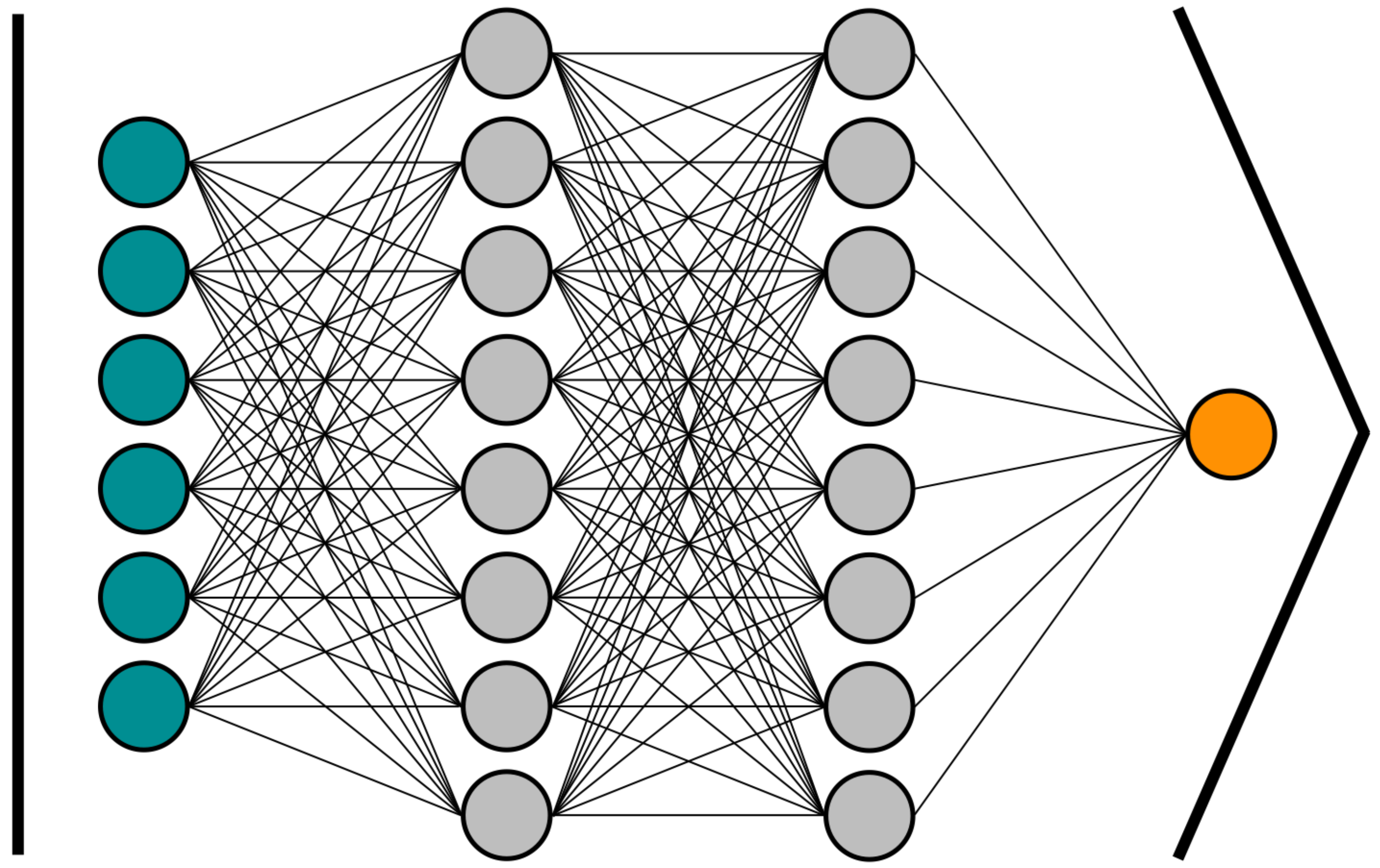
Collaborators

• Argonne National Laboratory:	**Alessandro Lovato** **Bryce Fore**
• École Polytechnique Fédérale de Lausanne:	**Giuseppe Carleo** **Gabriel Pescia** **Jannes Nys**
• Los Alamos National Laboratory:	**Stefano Gandolfi**
• Michigan State University / Facility for Rare Isotope Beams:	**Morten Hjorth-Jensen**

- See our preprint: **“Neural-network quantum states for ultra-cold Fermi gases”**
  arXiv:2305.08831

Overview

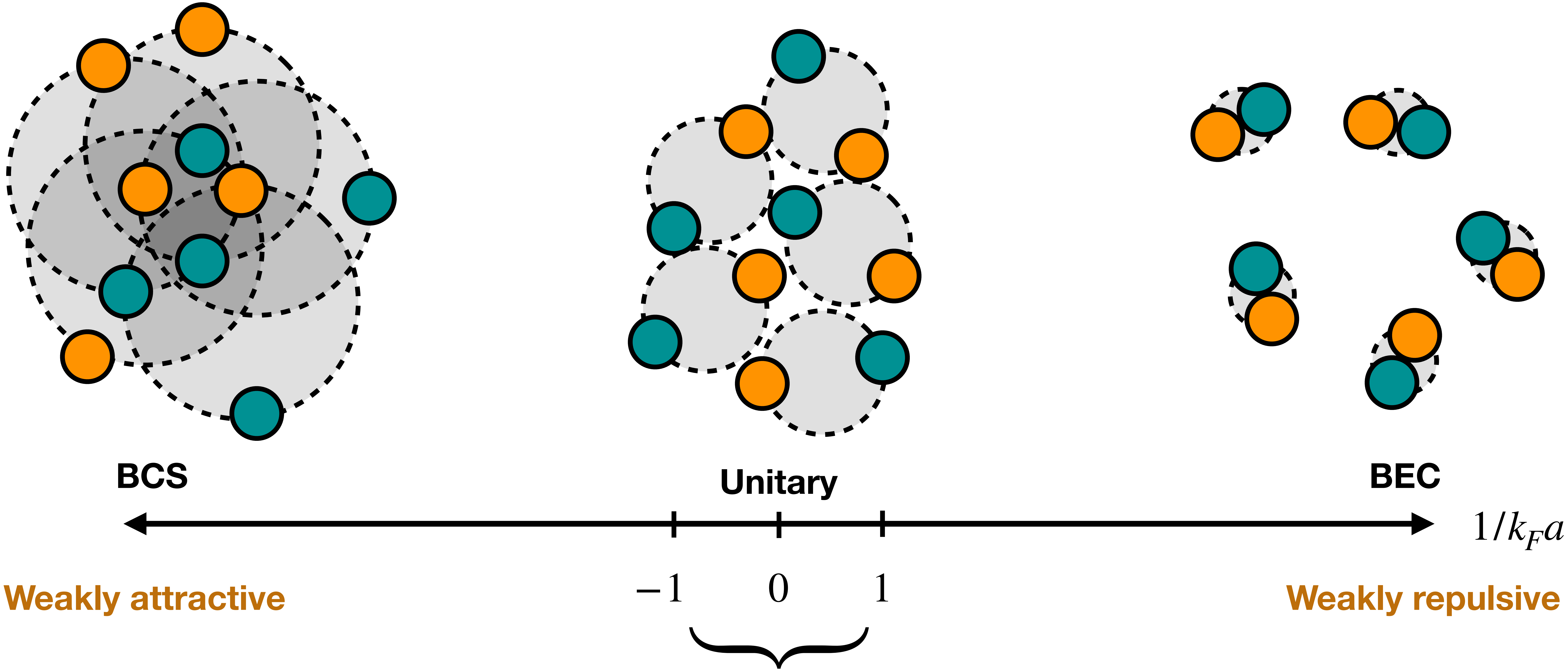
- Ultracold Fermi gases
- Slater determinants, BCS wave functions, and Pfaffians…
- Pfaffian-Jastrow neural-network quantum state
- Backflow correlations with message-passing neural networks
- Results
    - Ground-state energies
    - Opposite-spin pair distributions
    - Pairing gap
- Conclusions and perspectives



Ultracold Fermi gases

- Dilute, two-component fermions at zero temperature
- Behavior mainly governed by s-wave scattering length a and effective range r_e
- Highly non-perturbative, short-range, attractive interaction
- Can be created in the laboratory with variable scattering length a
  - $a < 0$ BCS regime of long-range Cooper pairs
  - $a > 0$ BEC regime of tightly-bound dimers
  - $|a| \to \infty$ Unitary limit (universal)
- Relevant for understanding superfluidity in fermionic systems, dilute neutron star matter, and development of many-body methods

BCS-BEC Crossover



BCS

Unitary

BEC

$1/k_F a$

-1 0 1

Weakly attractive

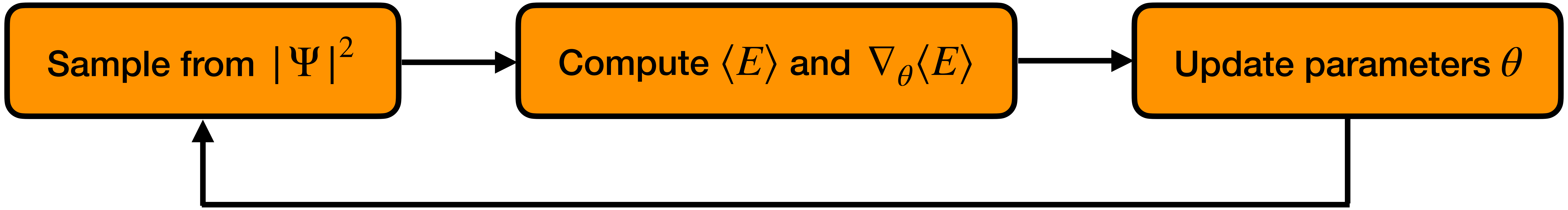
Weakly repulsive

Crossover region

Strongly interacting

Our Approach

- Simulate gas of N fermions in a periodic box of side length L
- Design a neural-network quantum state (NQS) that efficiently captures pairing and backflow correlations, while enforcing symmetries and boundary conditions
- Train NQS using variational Monte Carlo (VMC) method, i.e. minimize the energy $\langle E \rangle$


Sample from $|\Psi|^2$
Compute $\langle E \rangle$ and $\nabla_\theta \langle E \rangle$
Update parameters θ

- Compare to state-of-the-art Diffusion Monte Carlo (DMC) calculations
- Compare to similar NQS based on Slater determinants, with and without backflow

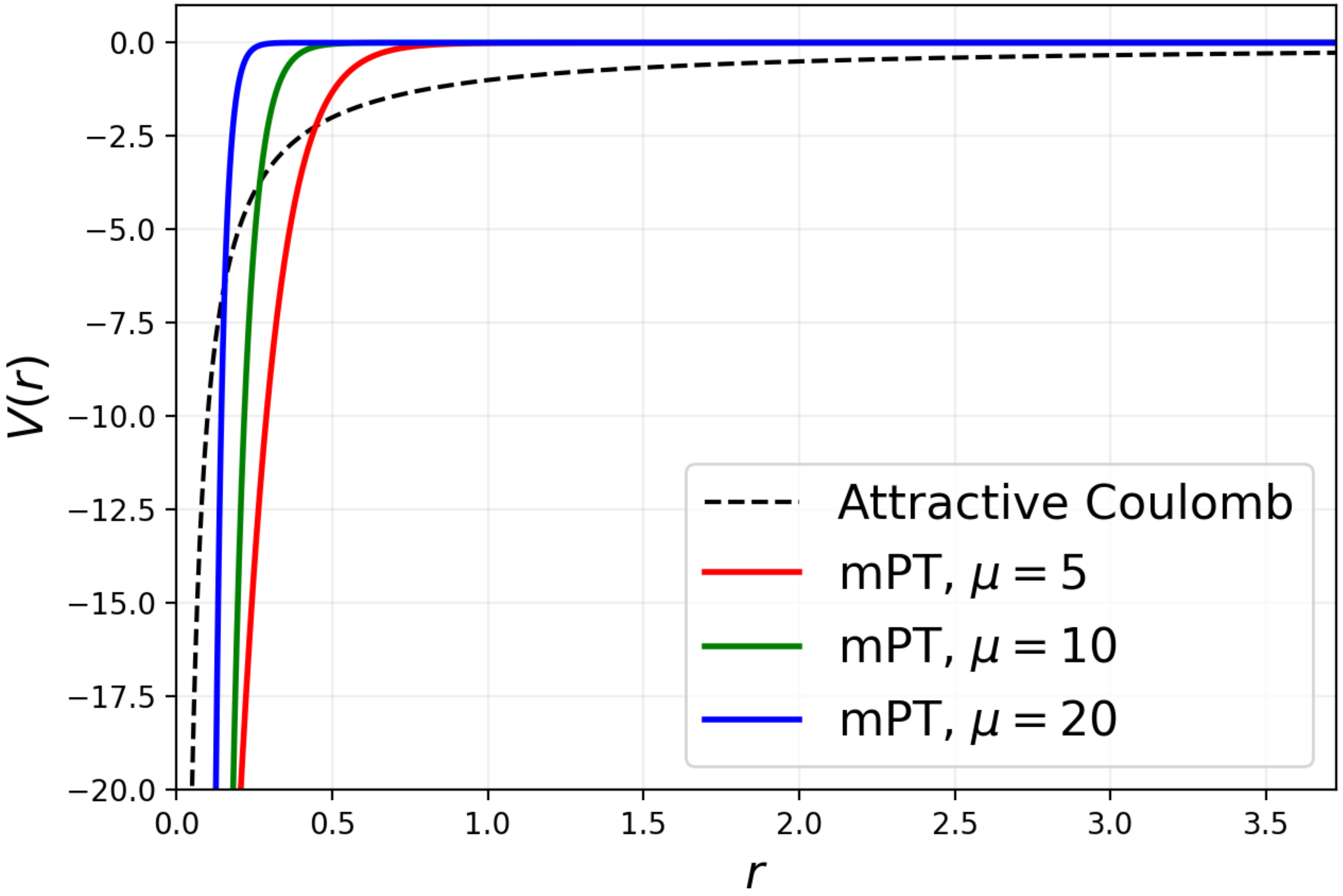
Pöschl-Teller Potential

- Regularized, short-range attraction between opposite-spin pairs

$$V_{ij} = (\delta_{s_i,s_j} - 1)\ v_0 \frac{2\hbar^2}{m} \frac{\mu^2}{\cosh^2(\mu r_{ij})}$$

- Provides exact solutions of two-body problem

- Other interaction potentials give similar results near unitarity as long as a and r_e are the same



Slater Determinants

- Wave function must be antisymmetric w.r.t. particle exchange

$$\Psi(X) = e^{J(X)}\Phi(X)$$

antisymmetric

symmetric

antisymmetric

$$X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$$
$$\mathbf{x}_i = (\mathbf{r}_i, s_i)$$

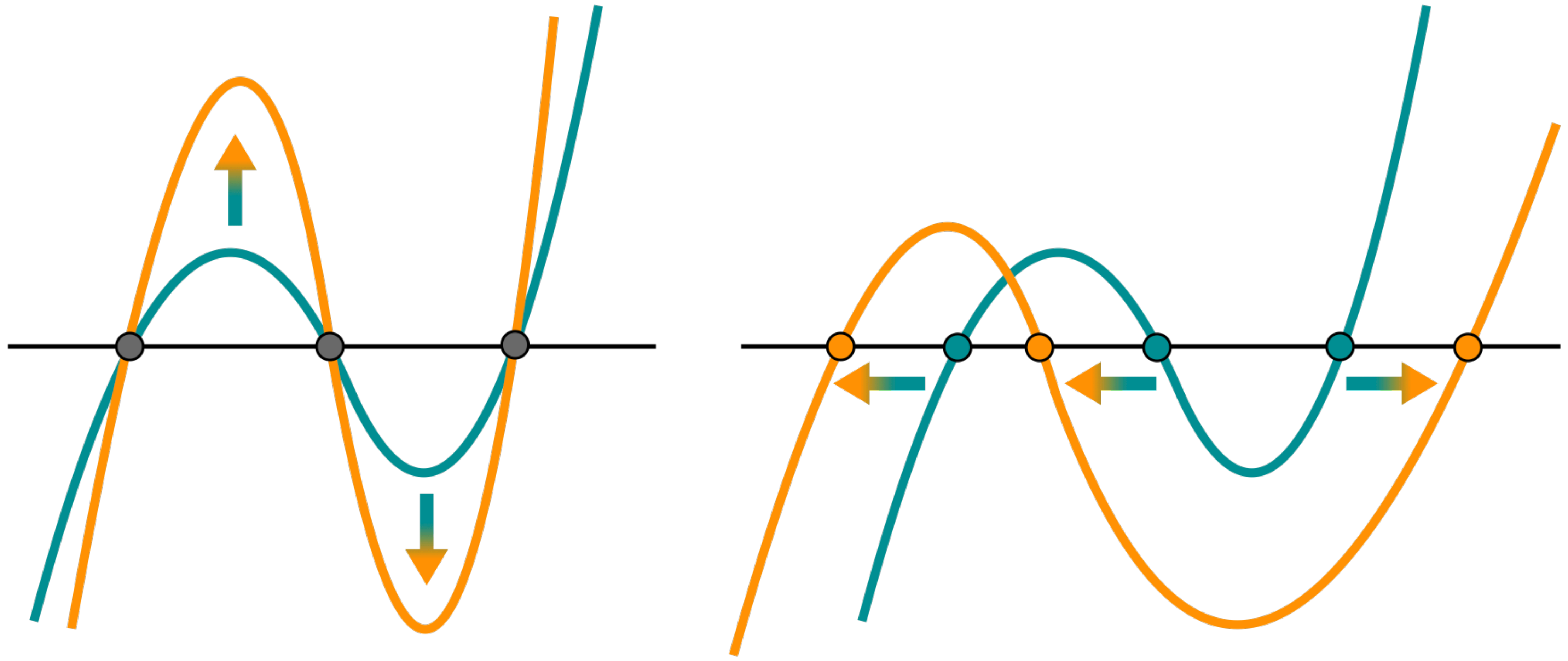
- Commonly use a Slater determinant of single-particle orbitals

$$\Phi_{SJ}(X) = \det \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \cdots & \phi_2(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\mathbf{x}_1) & \phi_N(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_N) \end{bmatrix}$$

- Fixed-node approximation with plane wave orbitals (SJ-PW): $\phi_\alpha(\mathbf{x}_i) = e^{i\mathbf{k}_\alpha \cdot \mathbf{r}_i}\delta_{s_\alpha, s_i}$

Backflow Transformations

- Symmetric, positive-definite Jastrow factor cannot change the nodes of the wave function



- Transform to a new basis that incorporates the influences from surrounding particles
- Transformation must be permutation equivariant to preserve antisymmetry

$$\mathbf{r}_i \mapsto \mathbf{r}_i + \mathbf{u}_i(X) + i\mathbf{v}_i(X)$$

$$|s_i\rangle \mapsto |\chi_i\rangle = \cos\left(\frac{\theta_i(X)}{2}\right)|s_i\rangle + \sin\left(\frac{\theta_i(X)}{2}\right)\sigma_i^x|s_i\rangle$$

BCS Wave Function

- Used for unpolarized systems with strong singlet pairing correlations

- Relies on separating spin-up and spin-down particles (cannot be used for nuclear systems)

$$\Phi_{BCS}(X) = \det \begin{bmatrix} \phi(\mathbf{x}_{1\uparrow}, \mathbf{x}_{1\downarrow}) & \phi(\mathbf{x}_{1\uparrow}, \mathbf{x}_{2\downarrow}) & \cdots & \phi(\mathbf{x}_{1\uparrow}, \mathbf{x}_{N/2\downarrow}) \\ \phi(\mathbf{x}_{2\uparrow}, \mathbf{x}_{1\downarrow}) & \phi(\mathbf{x}_{2\uparrow}, \mathbf{x}_{2\downarrow}) & \cdots & \phi(\mathbf{x}_{2\uparrow}, \mathbf{x}_{N/2\downarrow}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_{N/2\uparrow}, \mathbf{x}_{1\downarrow}) & \phi(\mathbf{x}_{N/2\uparrow}, \mathbf{x}_{2\downarrow}) & \cdots & \phi(\mathbf{x}_{N/2\uparrow}, \mathbf{x}_{N/2\downarrow}) \end{bmatrix}$$

- Can expand matrix to include unpaired orbitals for spin-polarized systems

Determinants vs. Pfaffians

Defined for $n \times n$ matrices

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$$\det(A^T) = \det(A)$$

$$\det(A)\det(B) = \det(AB)$$

Defined for $2n \times 2n$ skew-symmetric matrices

$$\operatorname{pf}(A) = \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{\sigma(2i-1),\sigma(2i)}$$

$$\operatorname{pf}(A^T) = (-1)^n \operatorname{pf}(A)$$

$$\operatorname{pf}(A)\operatorname{pf}(B) = \exp\left(\frac{1}{2}\operatorname{tr}\log(A^T B)\right)$$

$$\det(A) = \operatorname{pf}(A)^2$$

Pfaffian Wave Function

- Simplest and most general way to build an antisymmetrized product of pairing orbitals rather than single-particle orbitals

$$\Phi_{PJ}(X) = \mathrm{pf}\begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{x}_1) & \phi(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}_N) \\ \phi(\mathbf{x}_2, \mathbf{x}_1) & \phi(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \phi(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{x}_1) & \phi(\mathbf{x}_N, \mathbf{x}_2) & \cdots & \phi(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

- Commonly employed Pfaffian wave functions explicitly decompose pairing orbital into singlet and triplet components

- We take advantage of universal approximation theorem: $\phi(\mathbf{x}_i, \mathbf{x}_j) \equiv \nu(\mathbf{x}_i, \mathbf{x}_j) - \nu(\mathbf{x}_j, \mathbf{x}_i)$,
  where ν is a neural network.

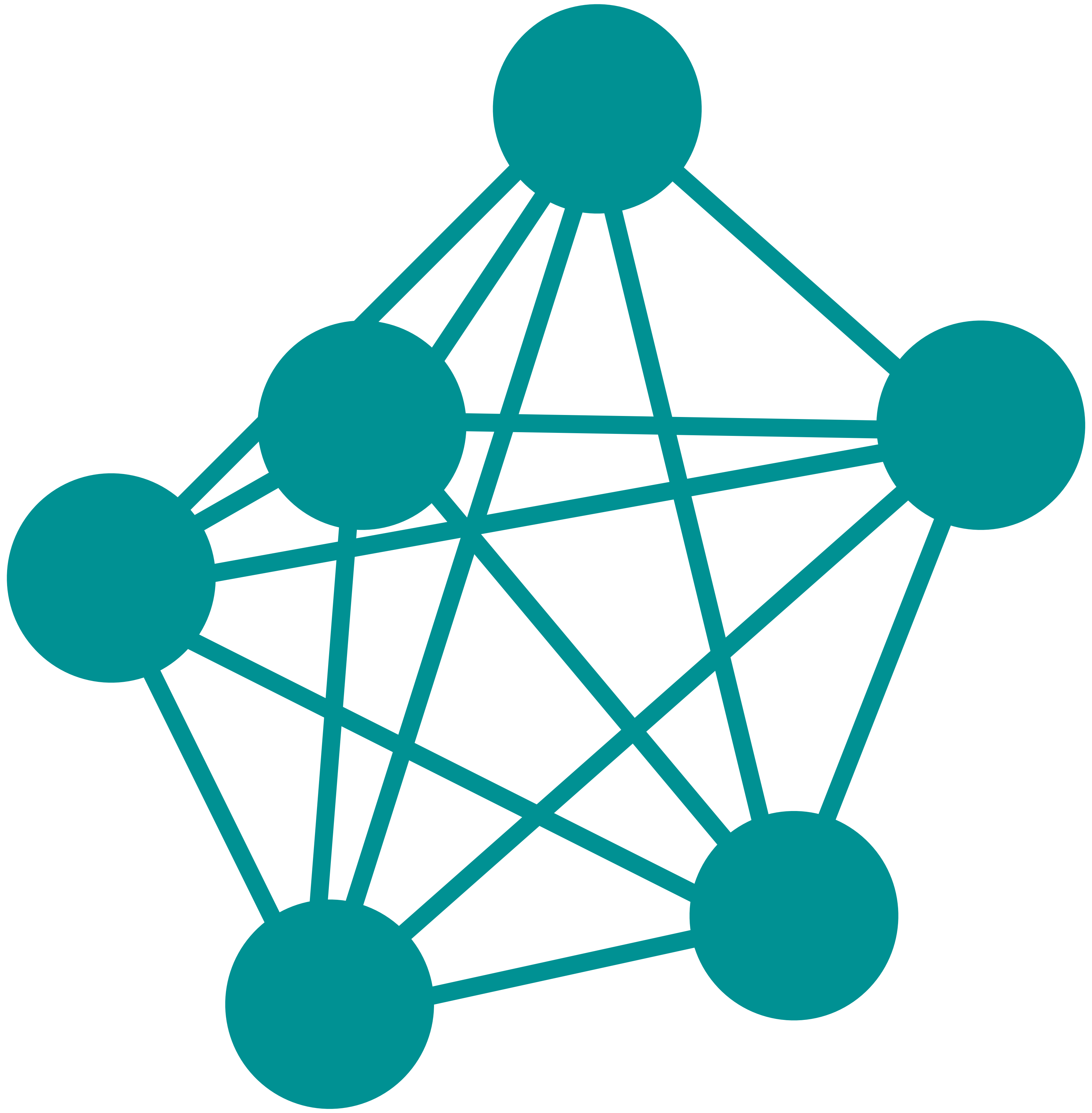
- Naturally encodes singlet and triplet pairing because ν takes spins as input

Message-Passing Neural Network

- Similar to the message-passing neural network used in our study of the homogenous electron gas (Pescia et al. arXiv:2305.07240)
- Permutation-equivariant graph neural network
- Iteratively builds correlations into new one-body and two-body features from original ones
- Skip connections help stabilize training and avoid vanishing gradients
- Visible nodes/one-body features: $\mathbf{v}_i = (s_i)$
- Visible edges/two-body features: $\mathbf{v}_{ij} = (r_{ij}, \mathbf{r}_{ij}, s_i \cdot s_j)$
- Preprocessing step: $\mathbf{h}_i^{(0)} = (\mathbf{v}_i, A\mathbf{v}_i)$

$$\mathbf{h}_{ij}^{(0)} = (\mathbf{v}_{ij}, B\mathbf{v}_{ij})$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

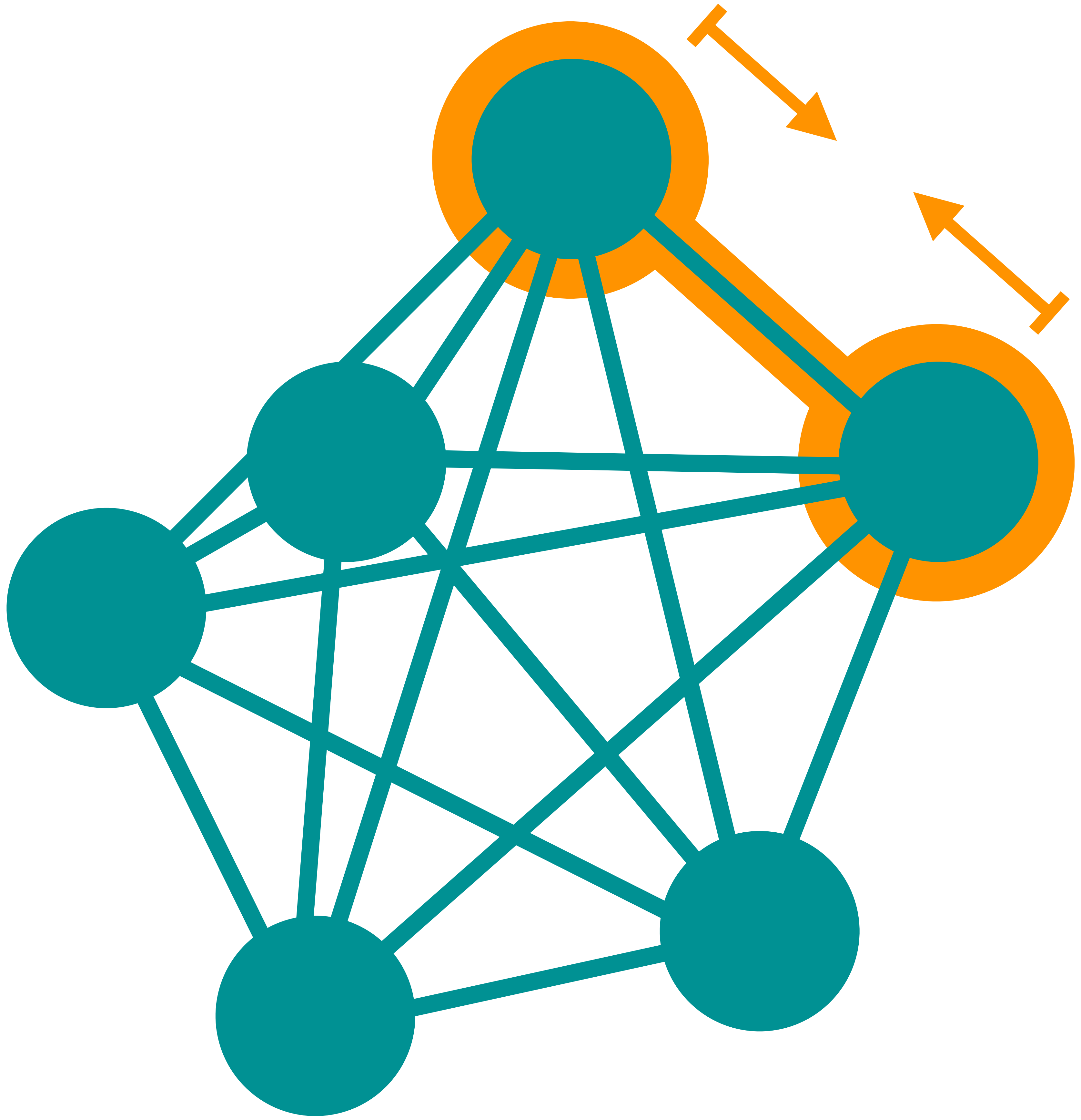
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t \left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)} \right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t \left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)} \right) \right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool} \left(\{ \mathbf{m}_{ij}^{(t)} | j \neq i \} \right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t \left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)} \right) \right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T:$

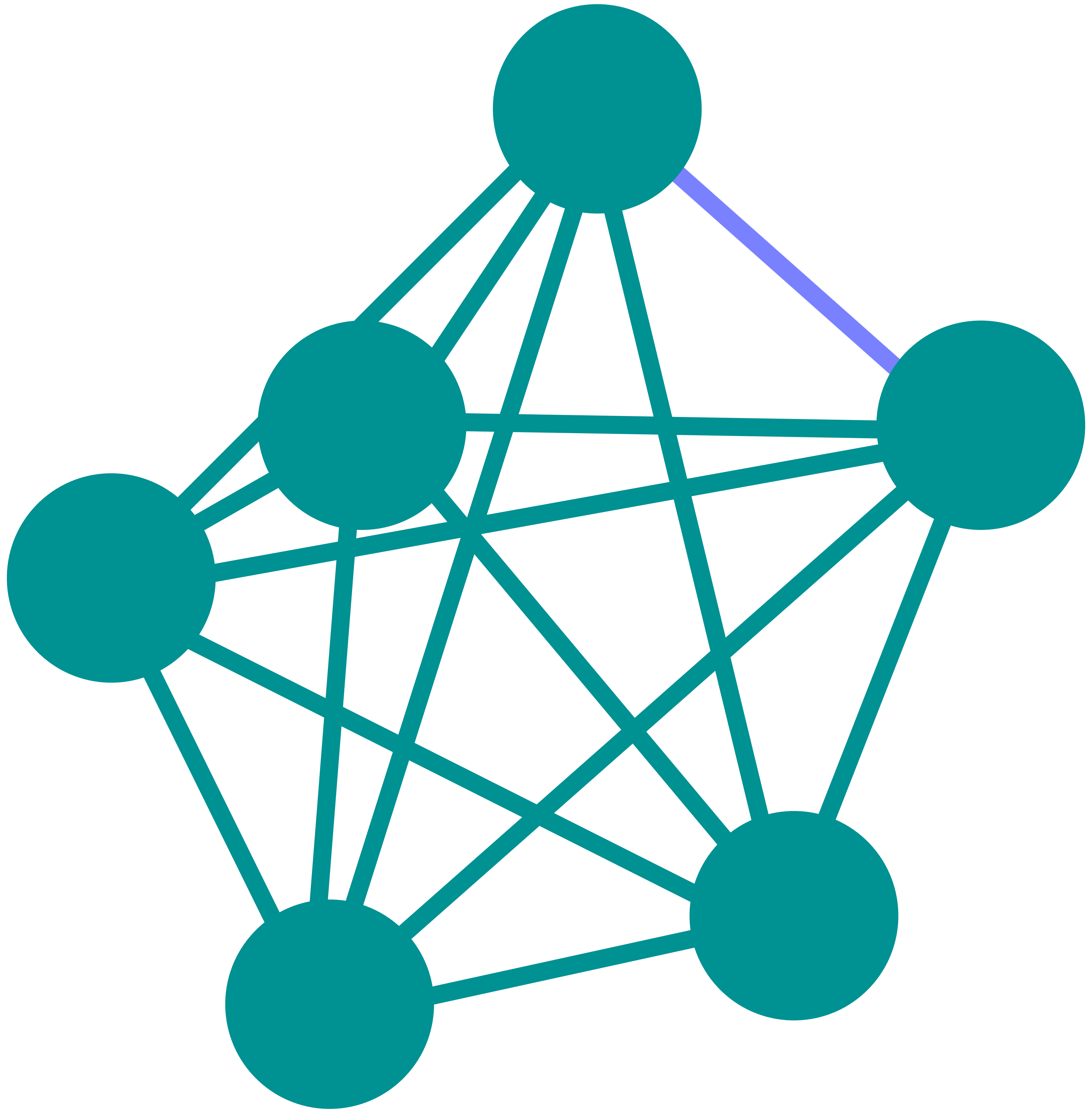
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

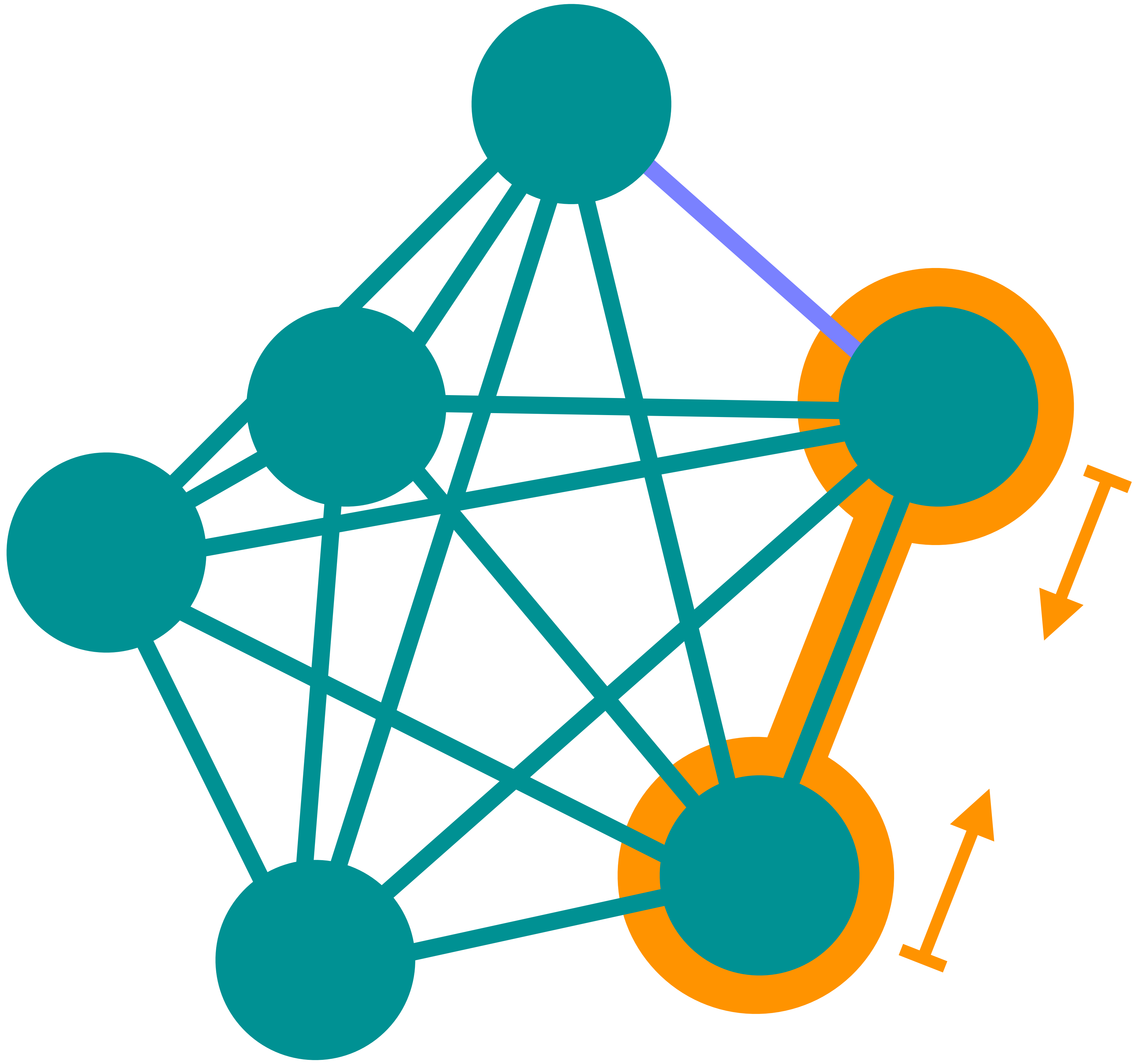
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

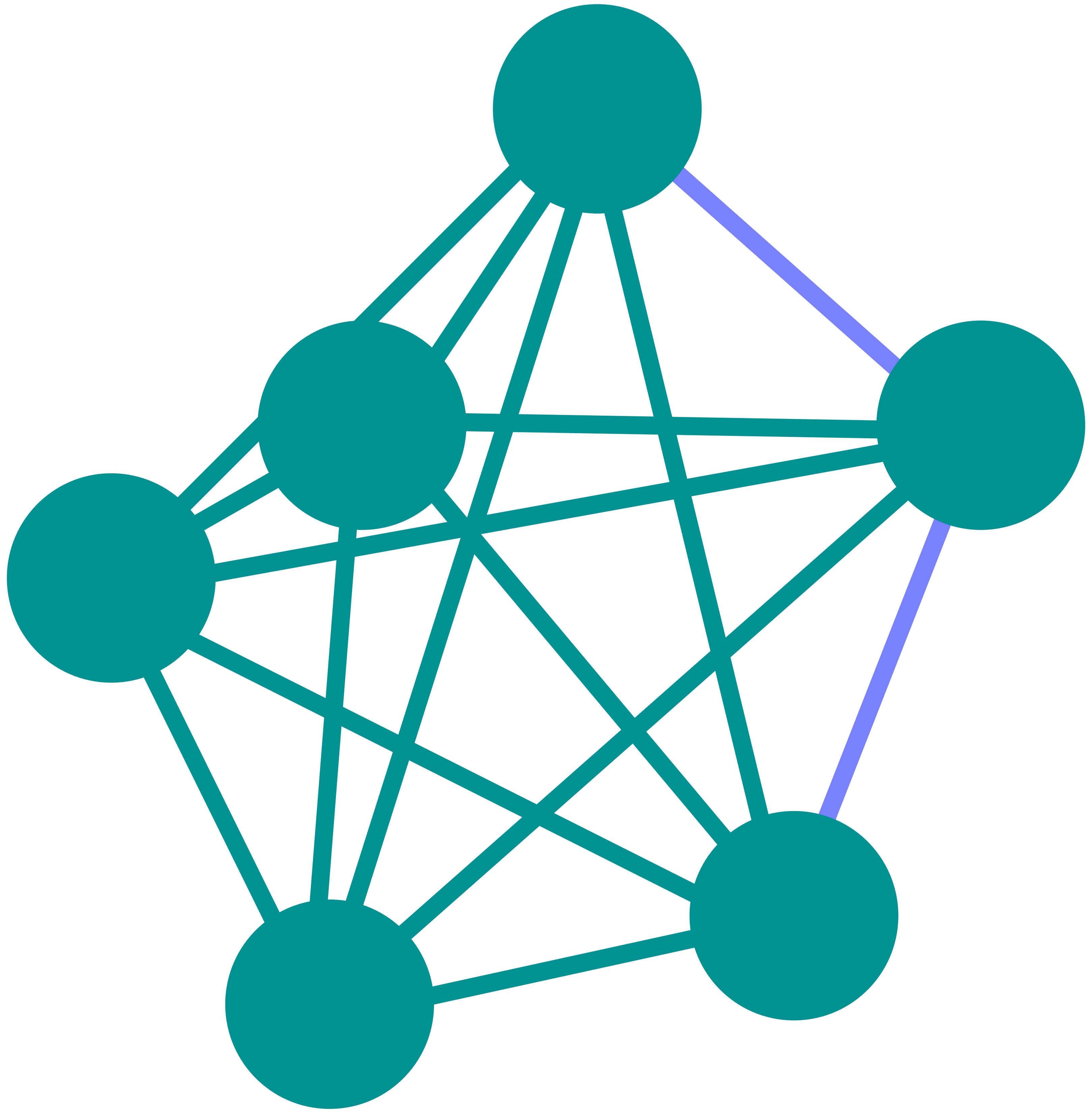
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

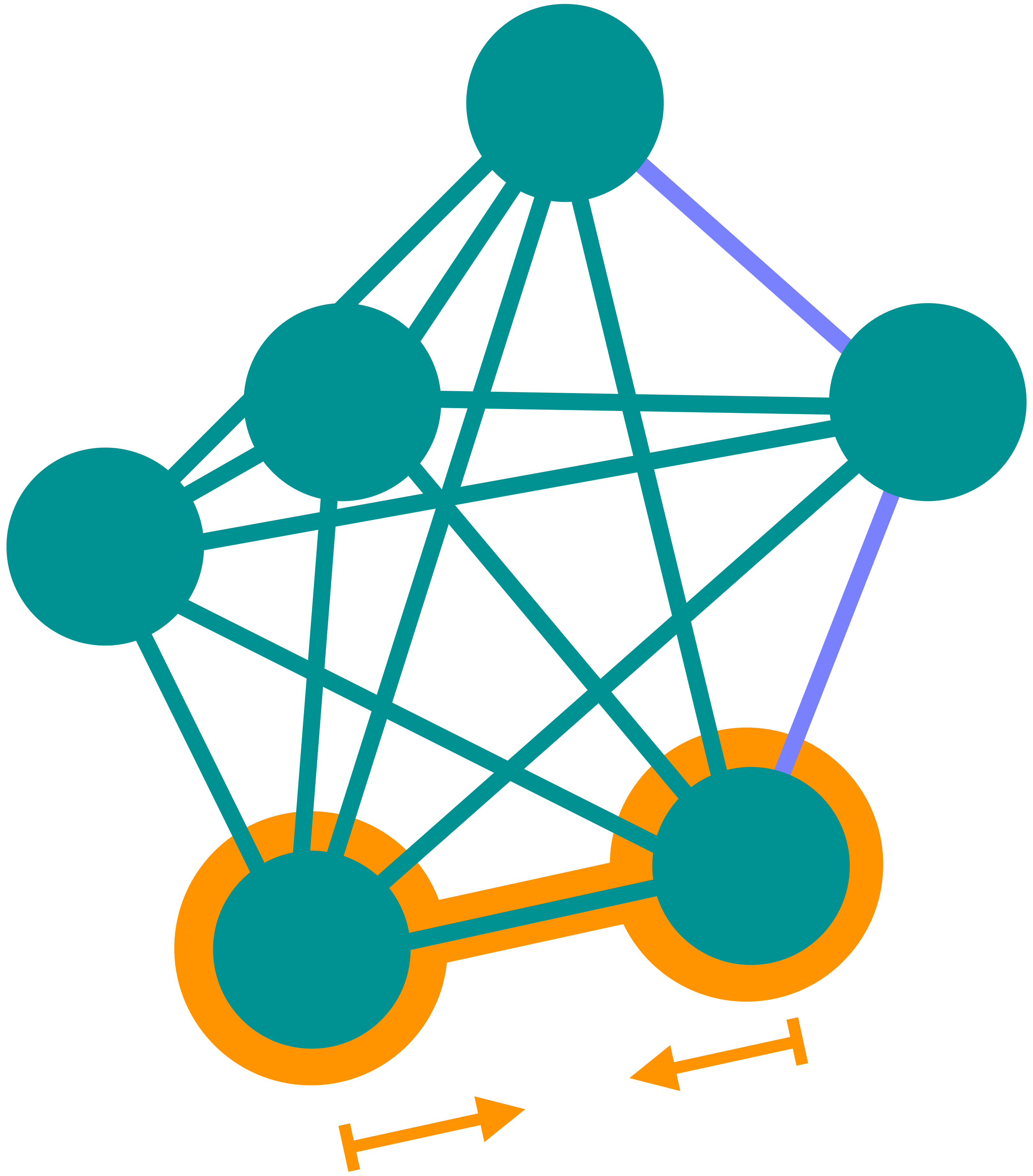
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \text{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

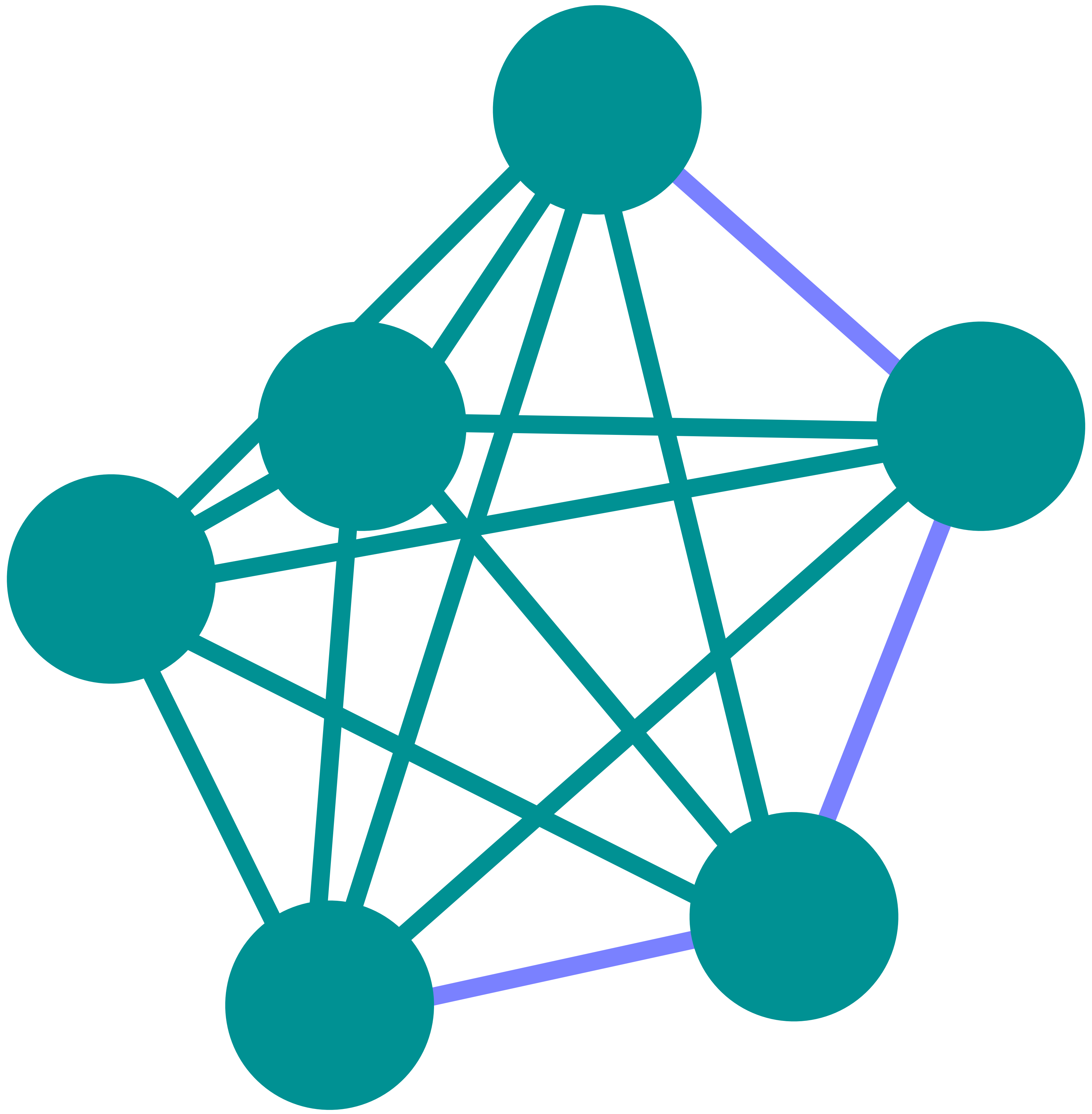
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \text{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

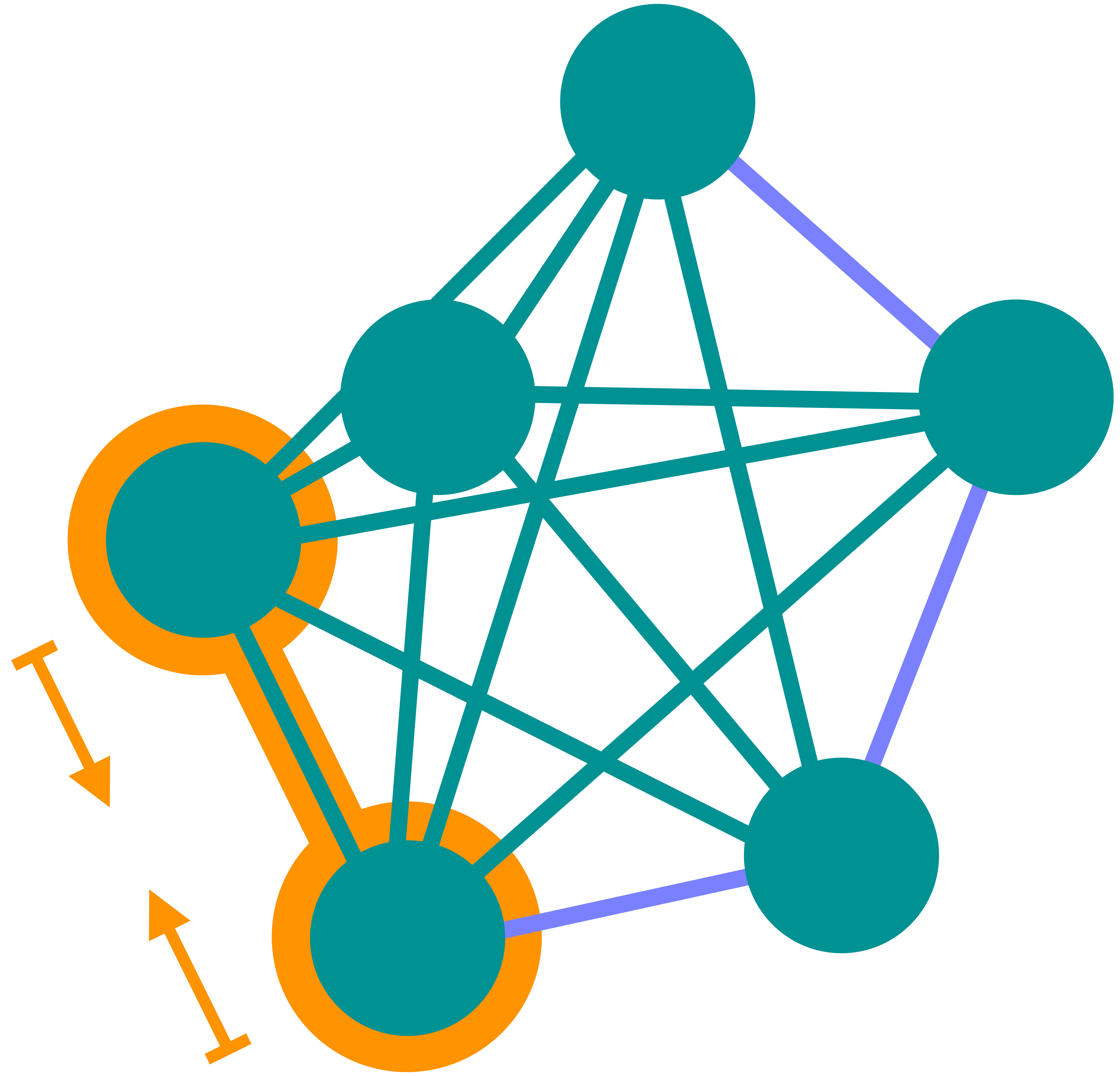
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

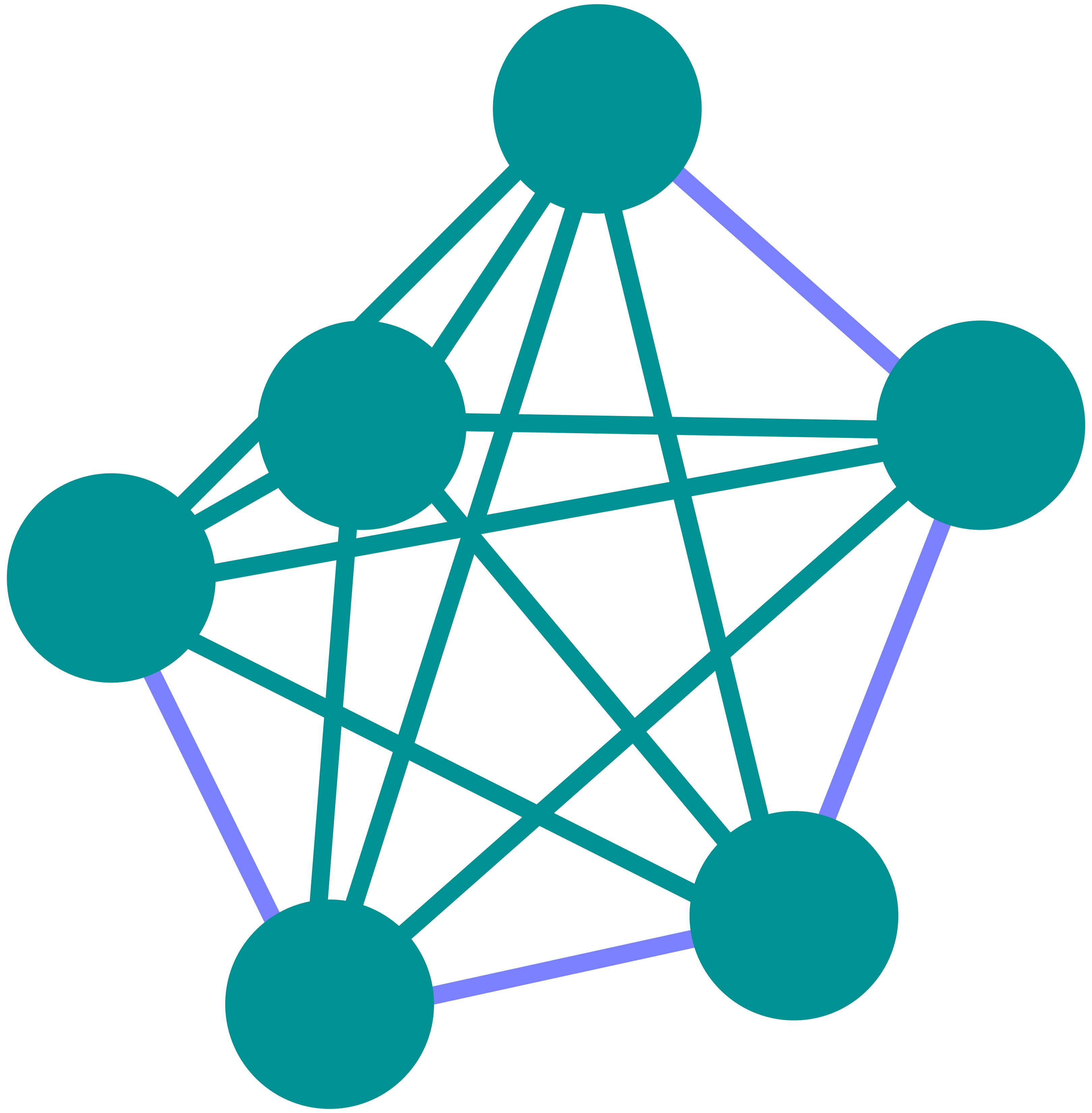
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \text{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

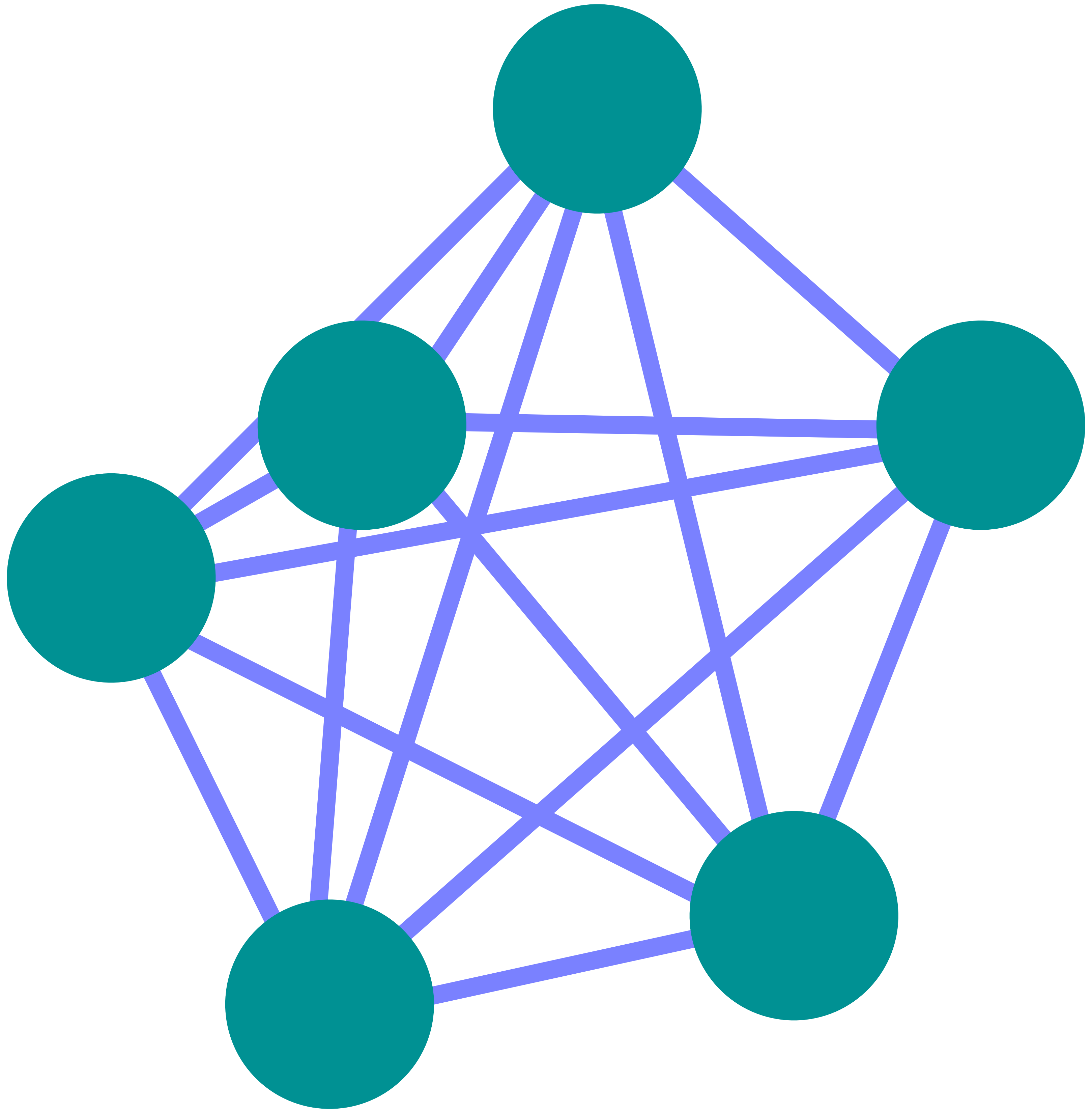
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \mid j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

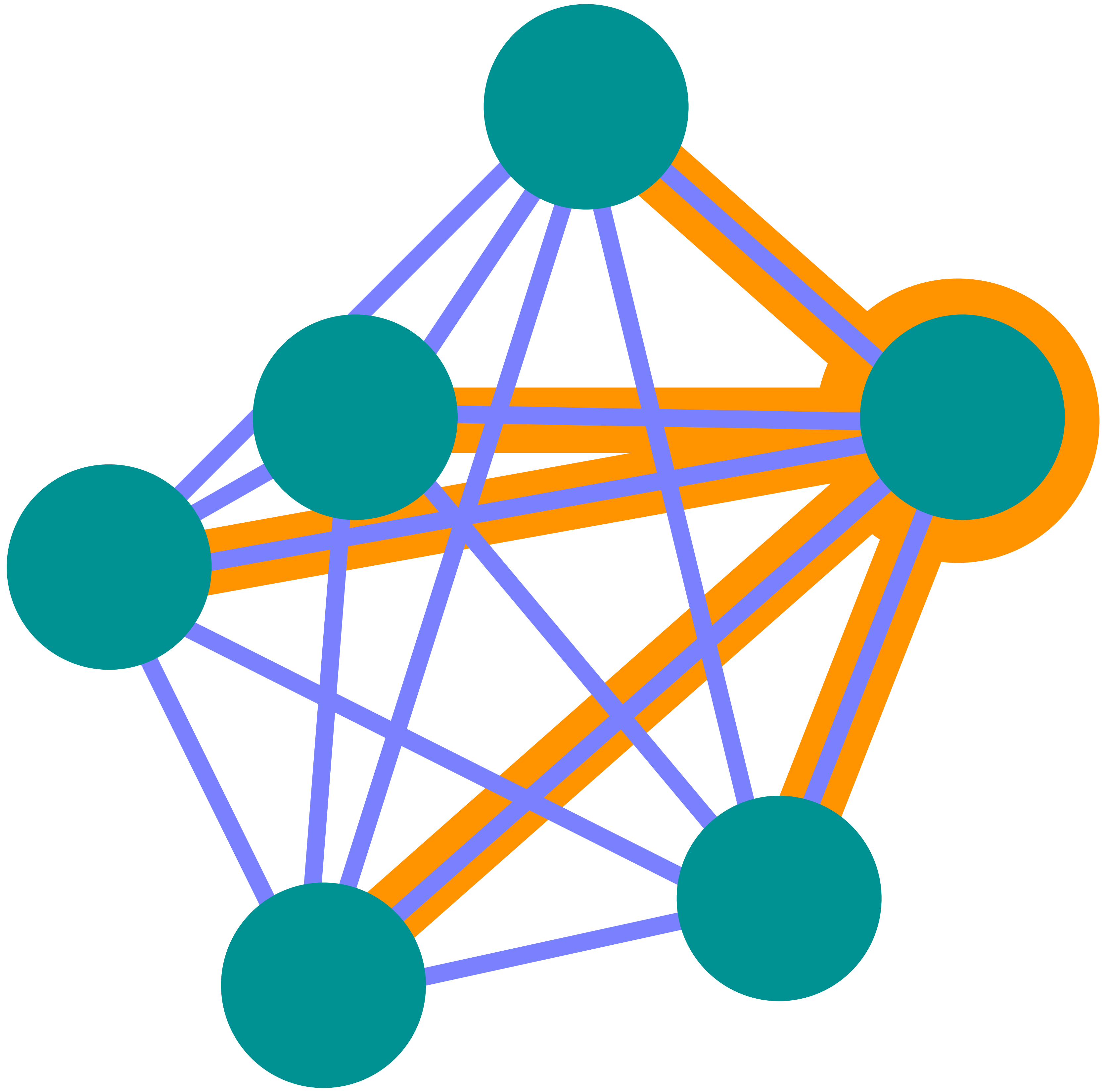
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

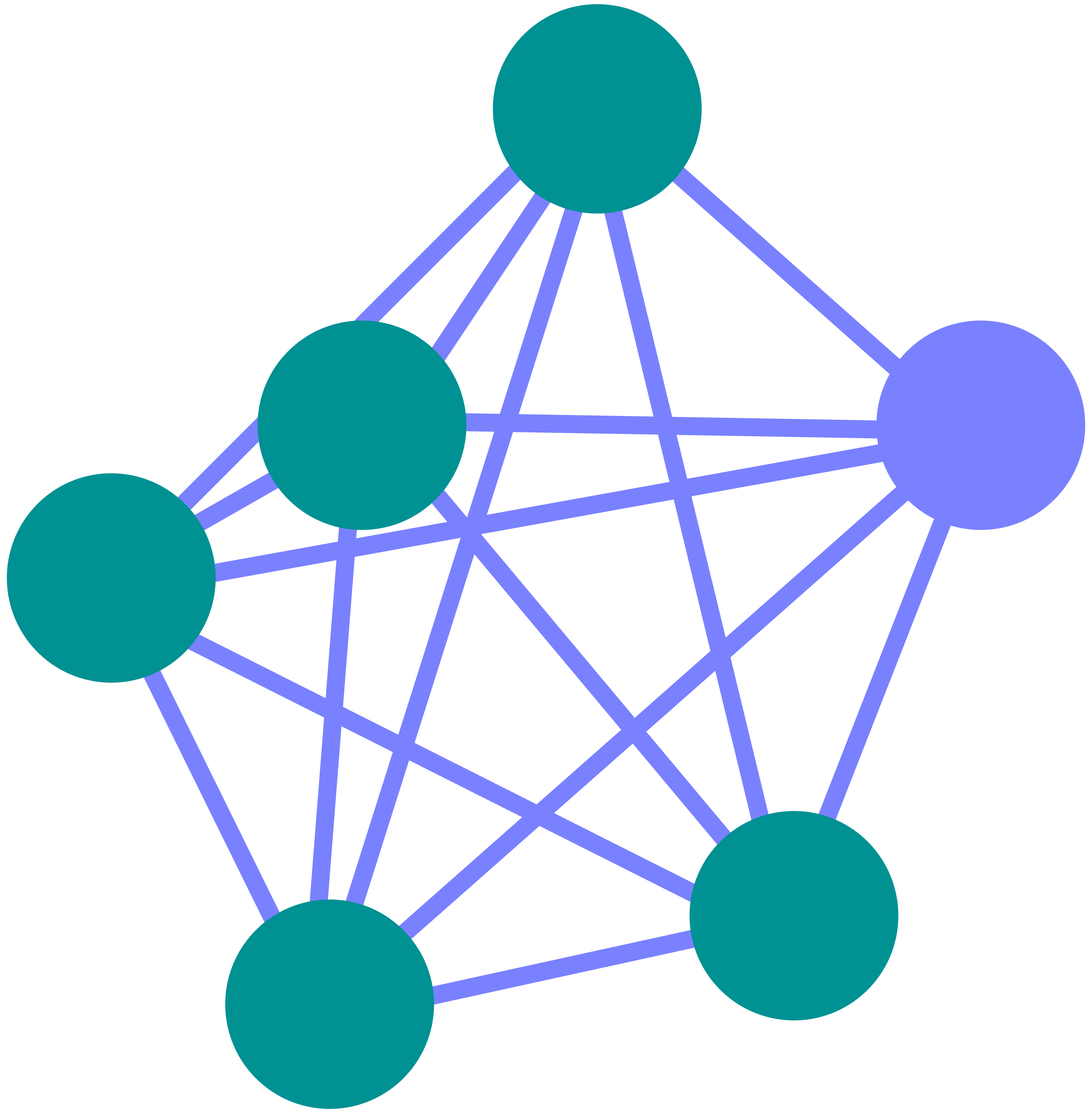
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)},\ \mathbf{h}_j^{(t-1)},\ \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij},\ \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \,|\, j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i,\ \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

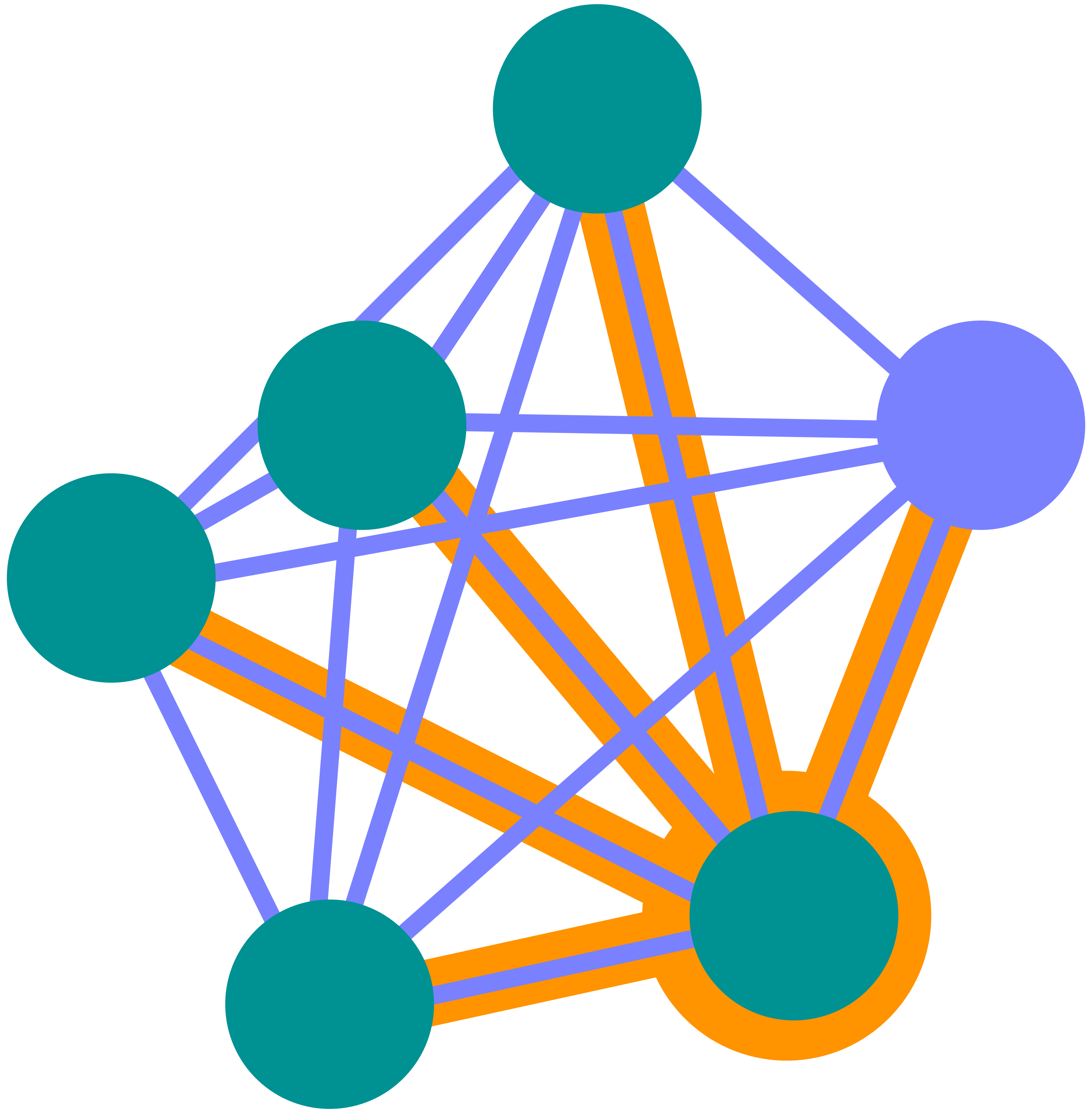
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \mid j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

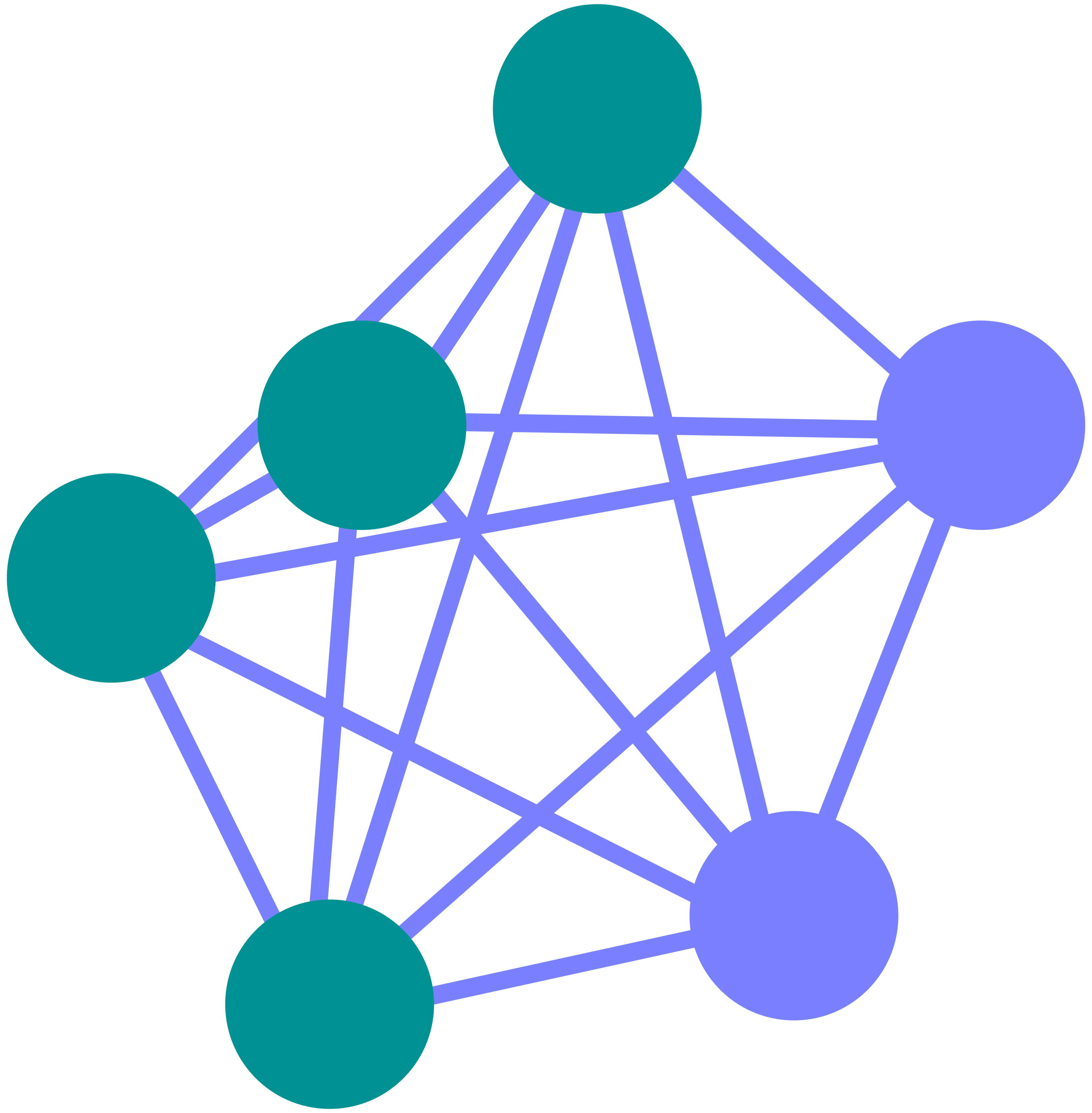
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \text{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

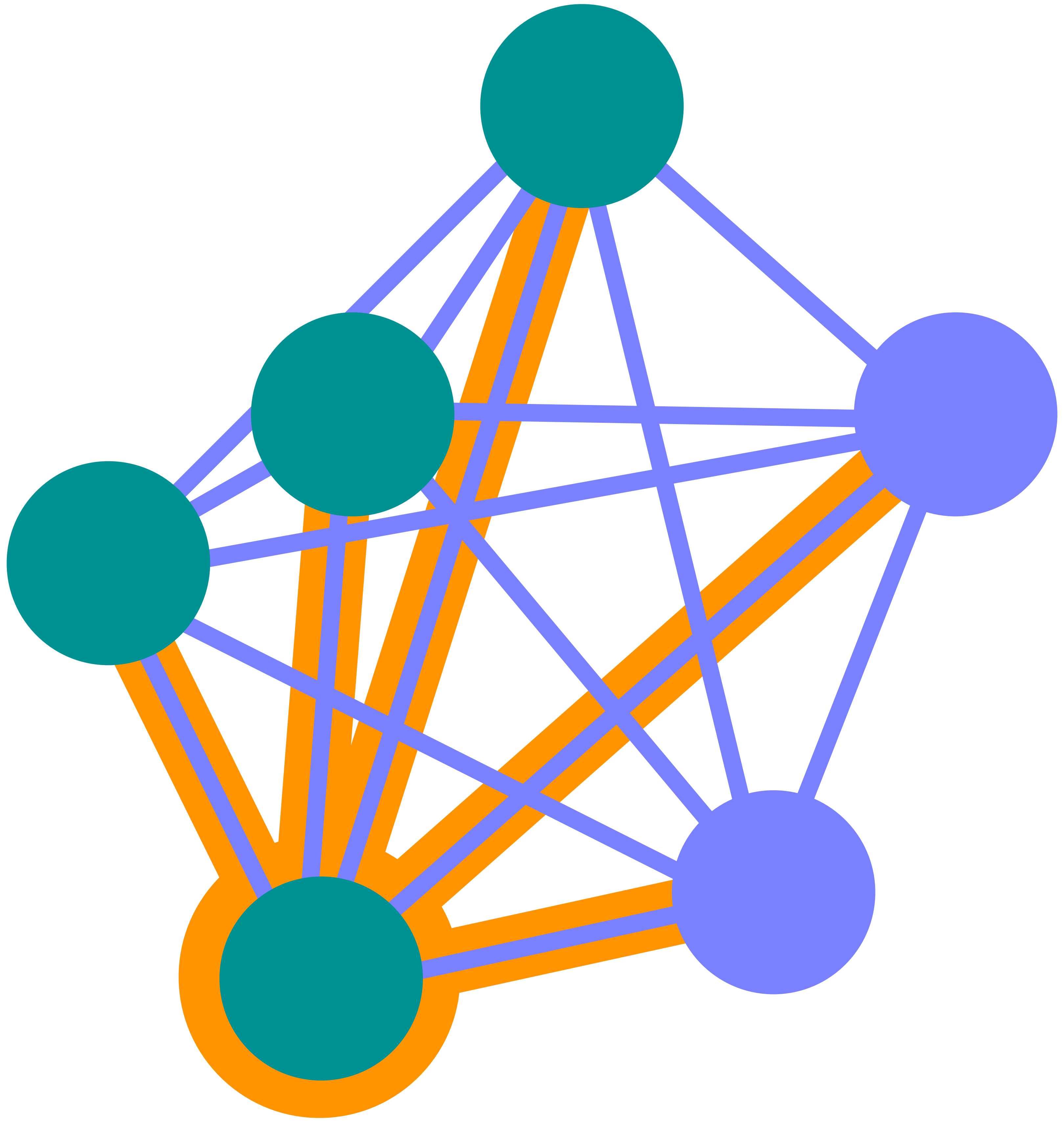
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \,|\, j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

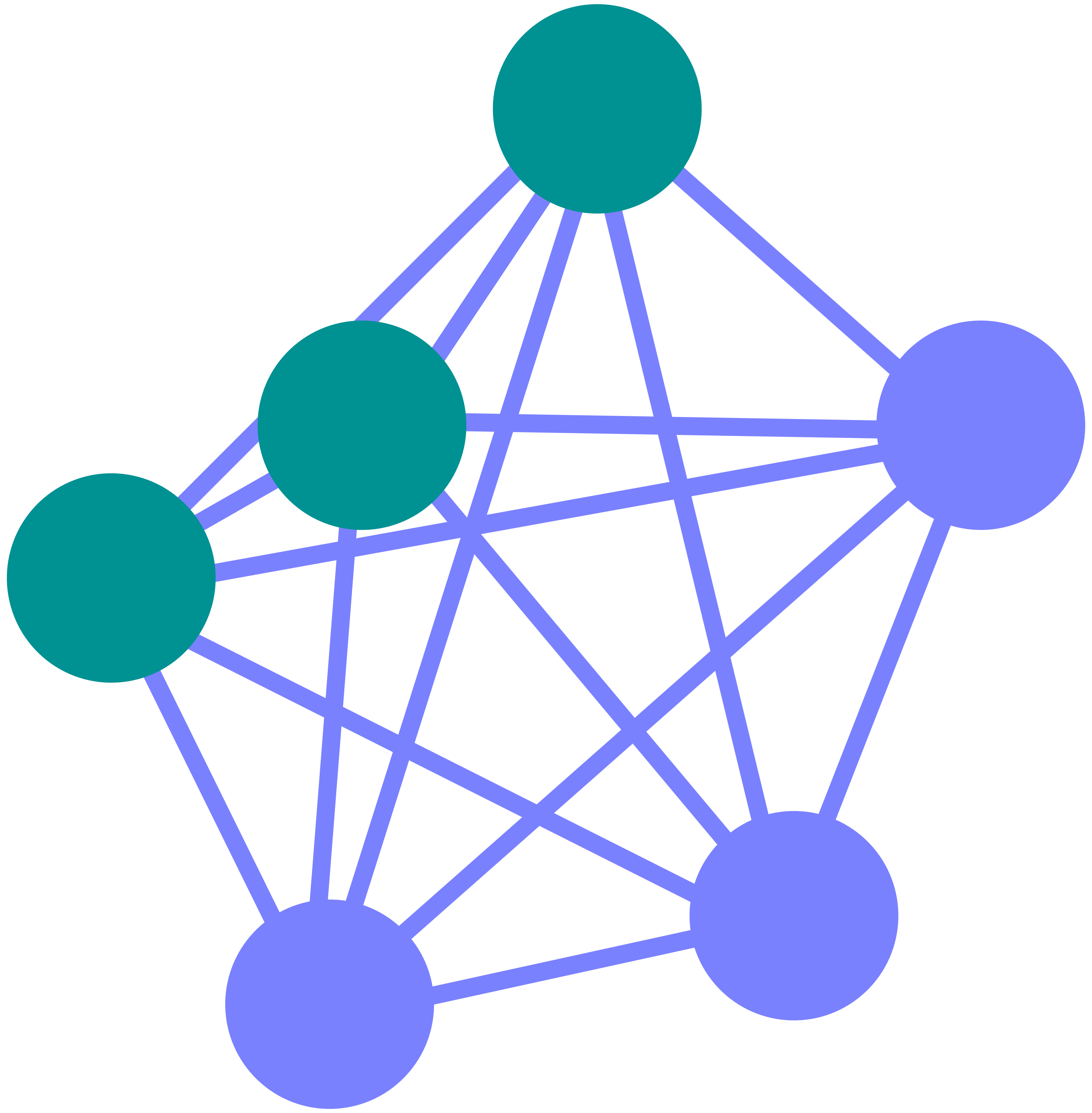
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \,|\, j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

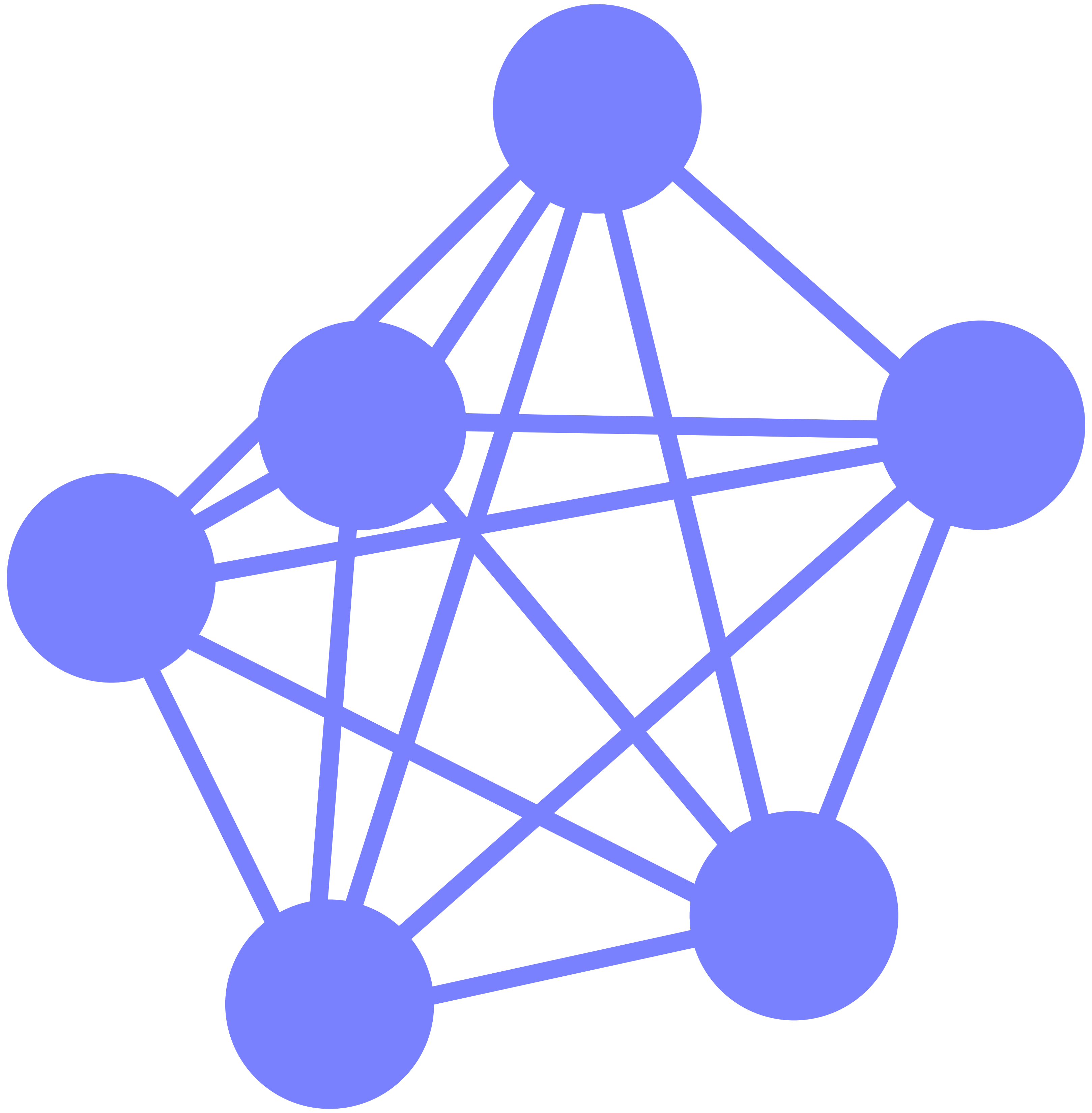
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \mid j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Message-Passing Neural Network



for $t = 1, \ldots, T$:

$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

$$\mathbf{h}_{ij}^{(t)} = \left(\mathbf{v}_{ij}, \mathbf{G}_t\left(\mathbf{h}_{ij}^{(t-1)}, \mathbf{m}_{ij}^{(t)}\right)\right)$$

$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} \,|\, j \neq i\}\right)$$

$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \mathbf{F}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

$\mathbf{v}_i$
$\mathbf{h}_i^{(0)}$
$\mathbf{h}_i^{(1)}$
...
$\mathbf{h}_i^{(T)}$
one-body stream
$\mathbf{m}_{ij}^{(1)}$
$\mathbf{m}_{ij}^{(2)}$
...
$\mathbf{m}_{ij}^{(T)}$
messages
$\mathbf{g}_{ij}$
$\mathbf{v}_{ij}$
$\mathbf{h}_{ij}^{(0)}$
$\mathbf{h}_{ij}^{(1)}$
...
$\mathbf{h}_{ij}^{(T)}$
two-body stream
layer 1
layer 2
layer T

Pfaffian-Jastrow-Backflow Ansatz

- The outputs of the message-passing neural network $\mathbf{g}_{ij} = (\mathbf{h}_i^{(T)}, \mathbf{h}_j^{(T)}, \mathbf{h}_{ij}^{(T)})$ are used as input to the trainable pairing orbital $\phi(\mathbf{x}_i, \mathbf{x}_j) \mapsto \phi(\mathbf{g}_{ij})$

- Periodicity:

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j \;\longmapsto\; \Big(\cos(2\pi\mathbf{r}_{ij}/L),\;\sin(2\pi\mathbf{r}_{ij}/L)\Big)$$

$$\|\mathbf{r}_{ij}\| \;\longmapsto\; \|\sin(2\pi\mathbf{r}_{ij}/L)\|$$

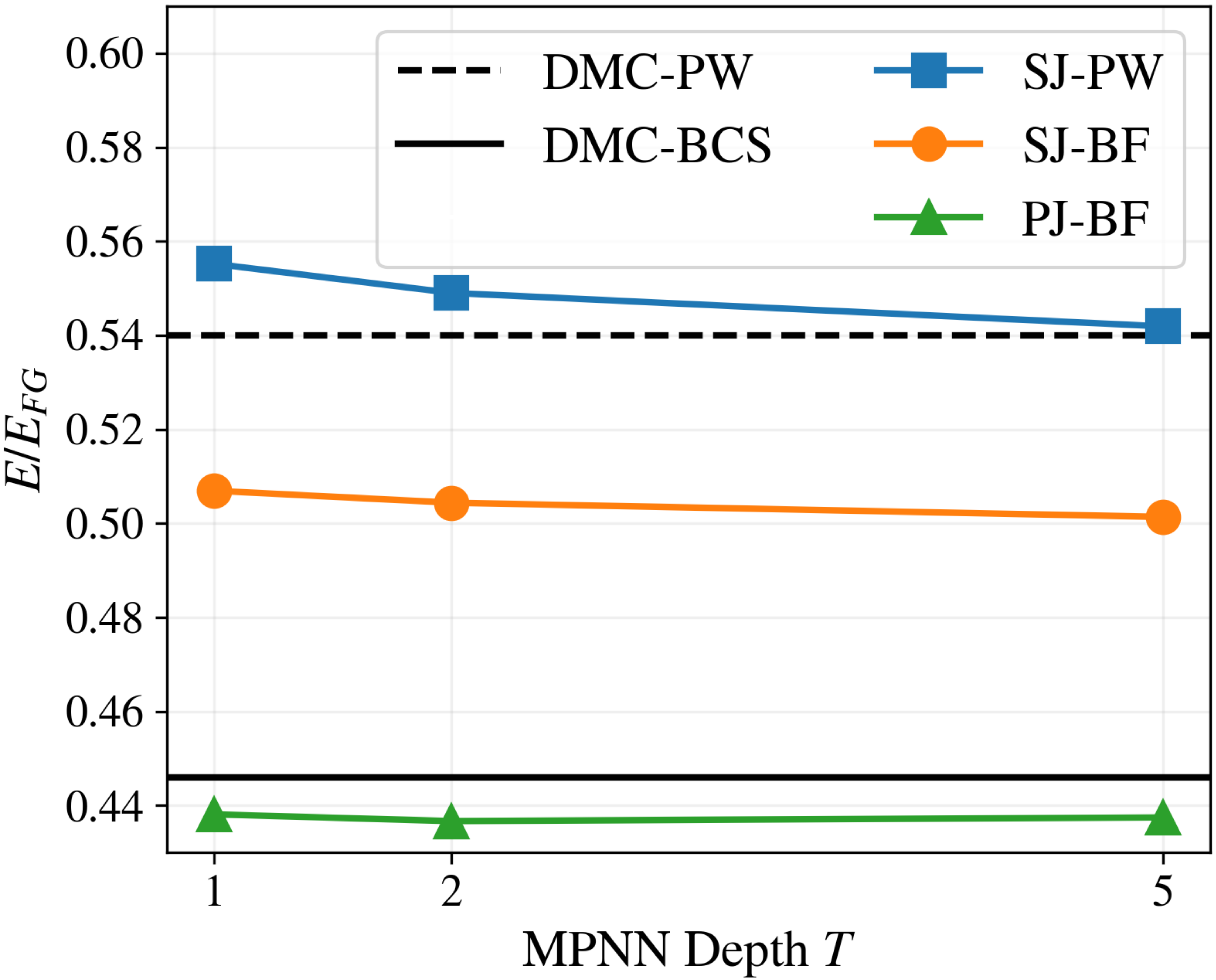
- Parity and time-reversal:

$$\Psi^P(R,S) = \Psi(R,S) + \Psi(-R,S)$$

$$\Psi^{PT}(R,S) = \Psi^P(R,S) + (-1)^{N/2}\Psi^P(R,-S)$$

- Positions are ignored to enforce translational invariance

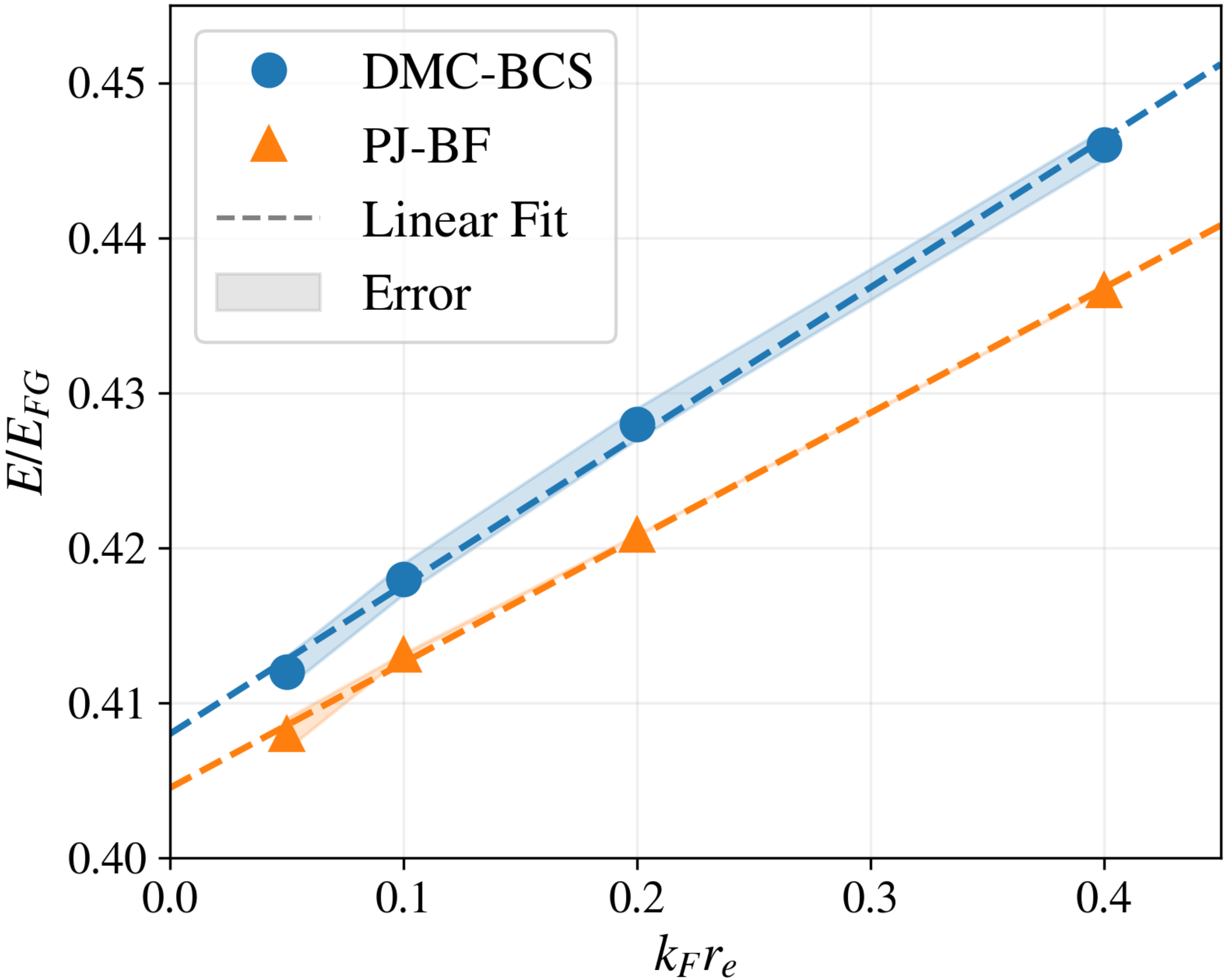
- Jastrow correlator constructed as a Deep Set over $\mathbf{g}_{ij}$



Initial comparison of NQS at unitarity

$k_F r_e = 0.4$

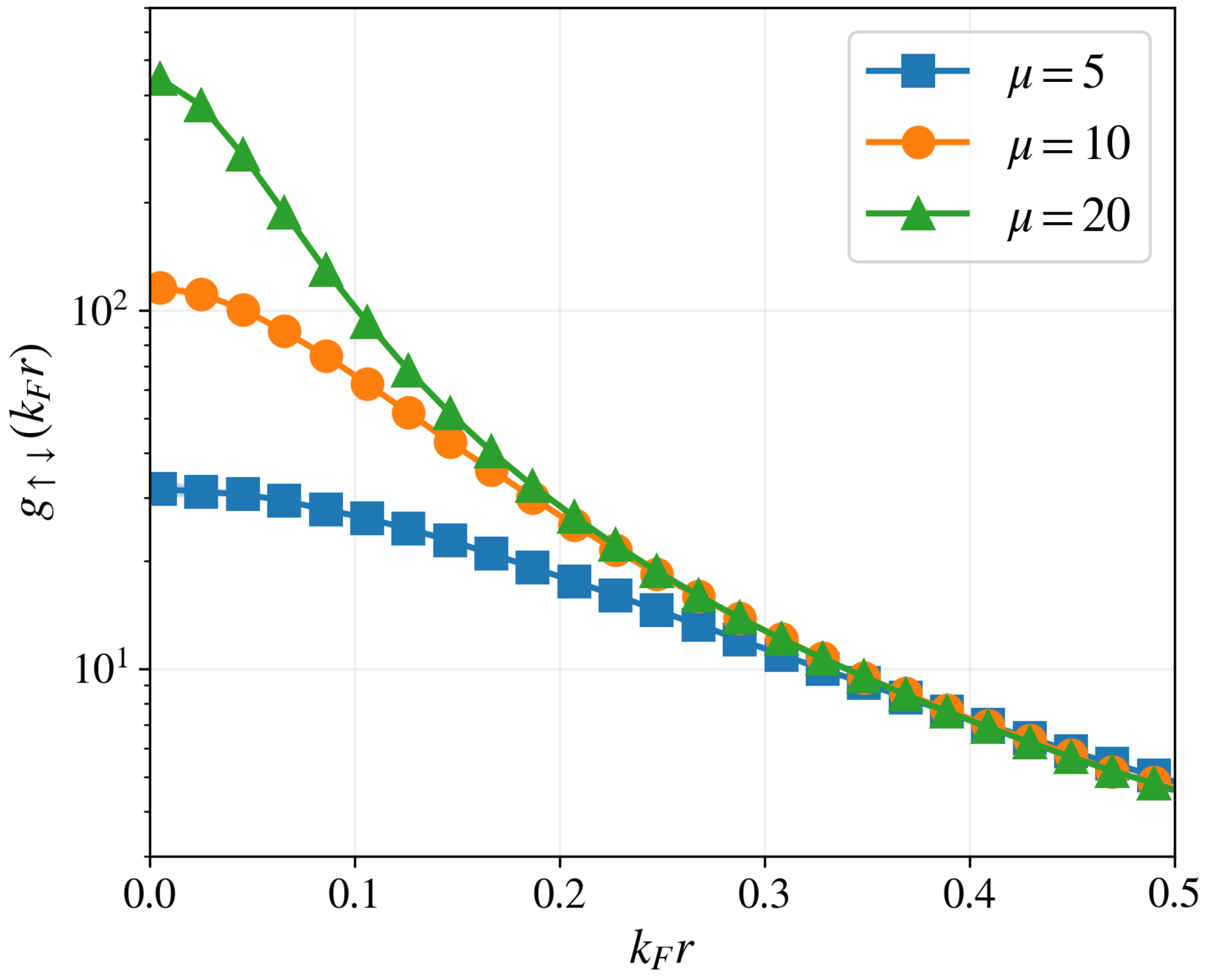
$1/ak_F = 0$



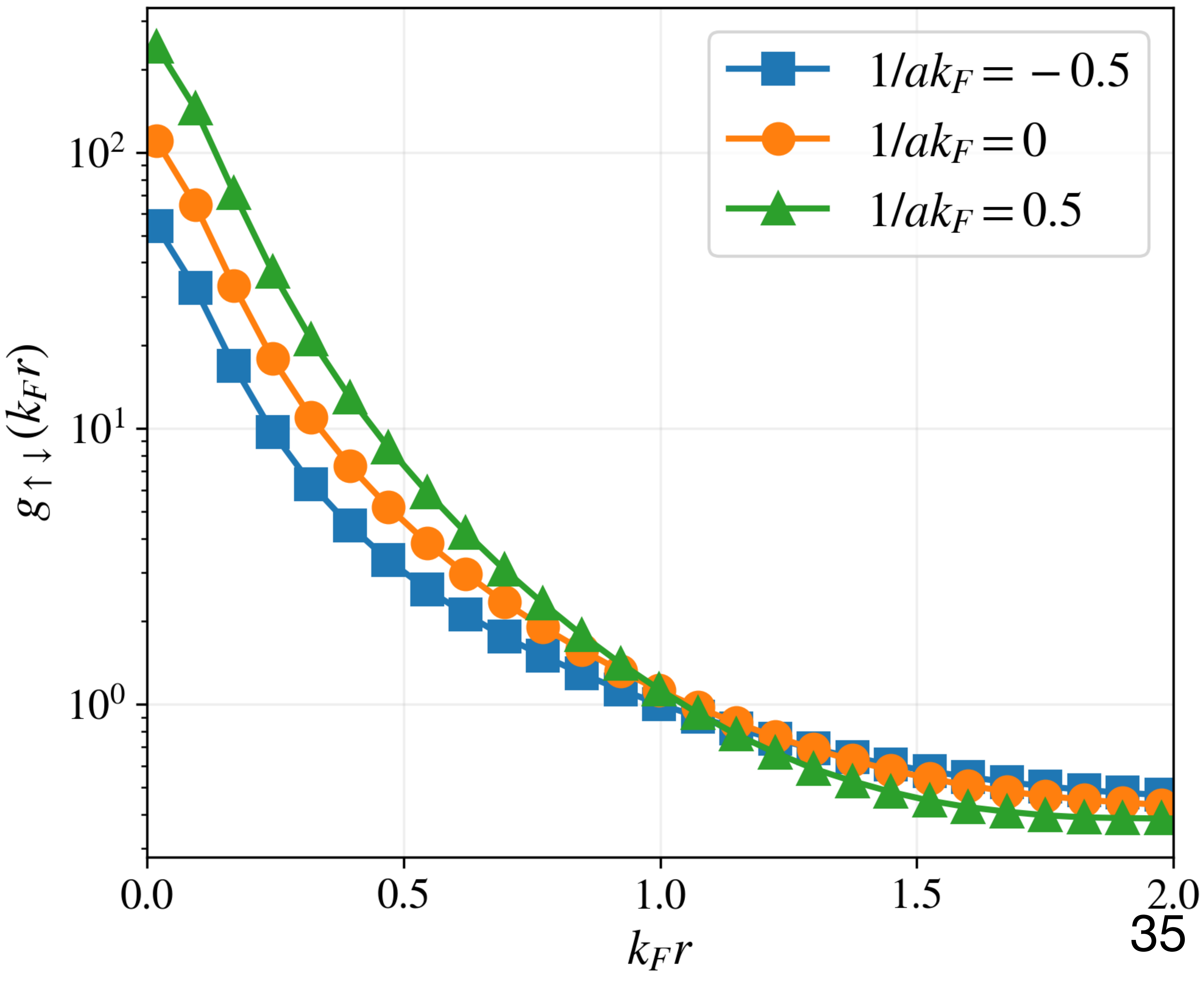
Extrapolation to zero $k_F r_e$

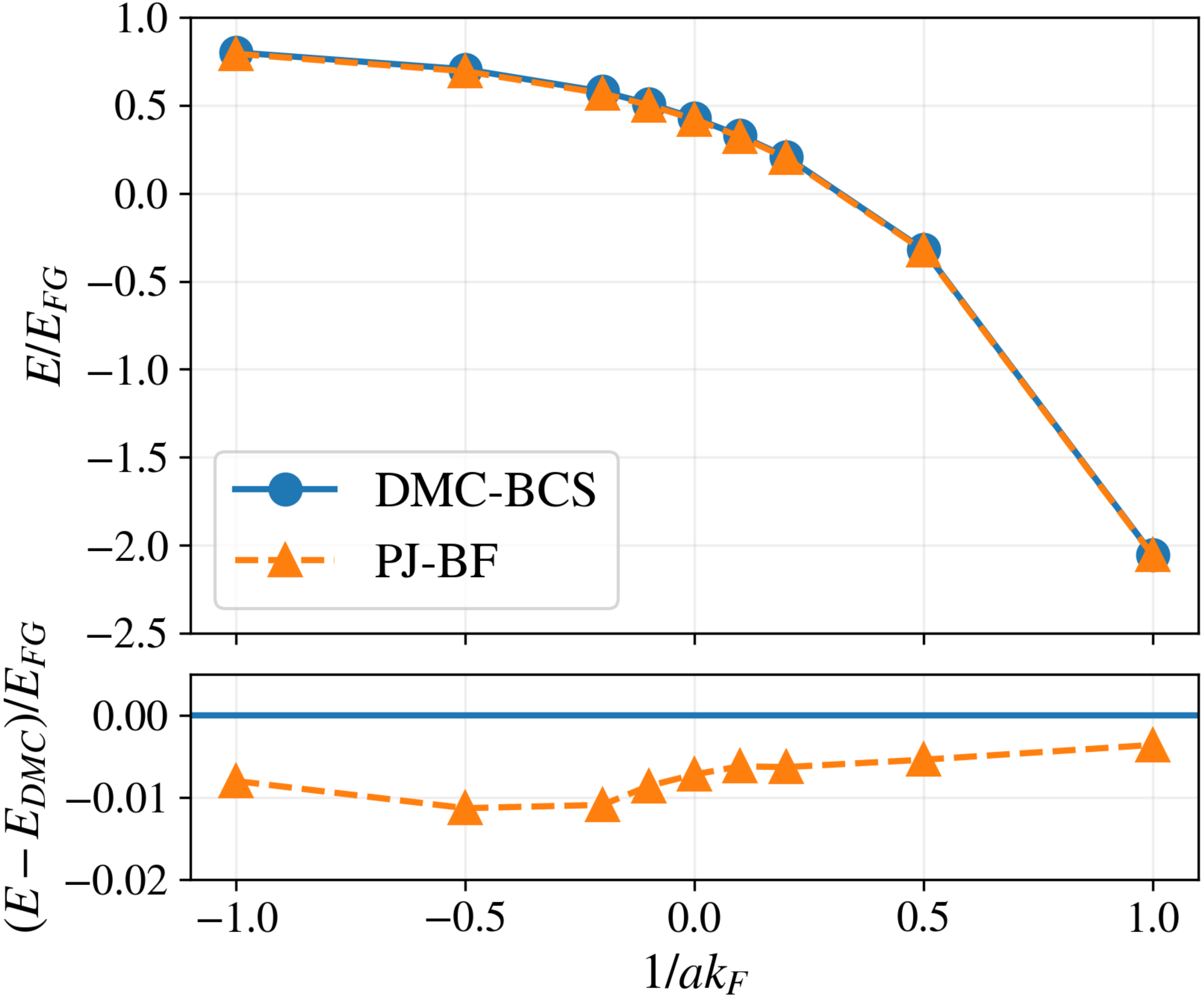
$$1/ak_F = 0$$

Different effective ranges $r_e = 2/\mu$ at unitarity $(1/ak_F = 0)$



Different scattering lengths near unitarity (fixed $r_e = 0.2$)





1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
E/E_{FG}
DMC-BCS
PJ-BF
0.00
−0.01
−0.02
$(E-E_{DMC})/E_{FG}$
−1.0
−0.5
0.0
0.5
1.0
$1/ak_F$

N-Independence

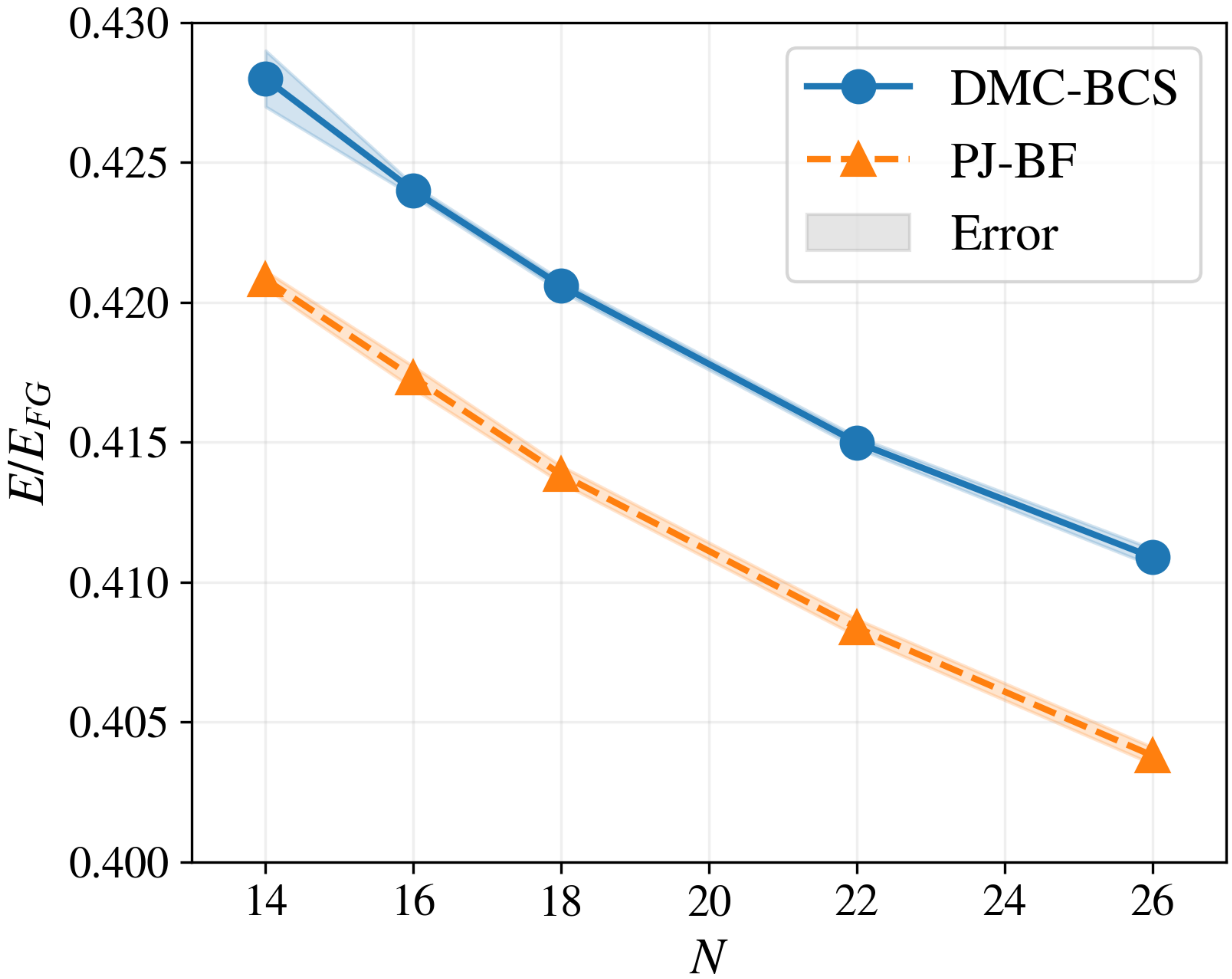
- Sizes of all trainable networks only depend on spatial dimension and chosen hidden layer dimensions

- Pfaffian NQS can be pretrained with smaller N to stabilize overall training

- Odd-N cases require one additional feedforward neural network to represent the unpaired single-particle orbital

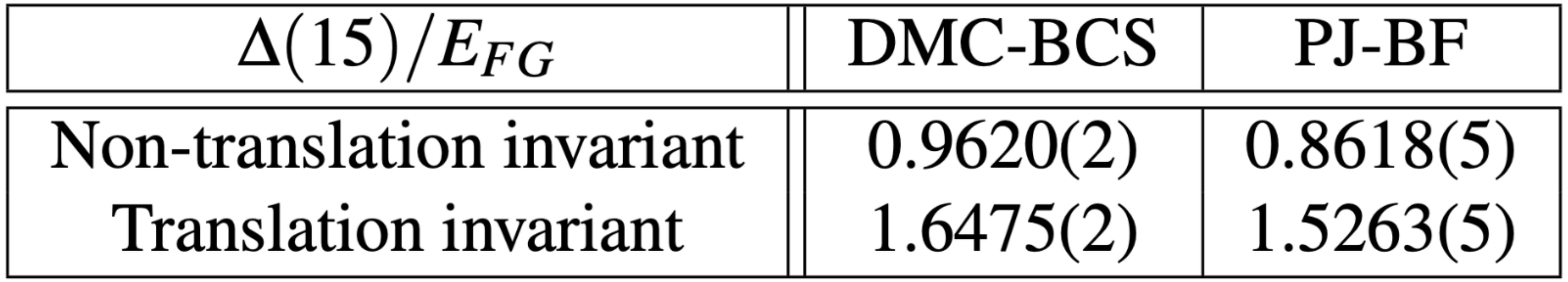
$$\Phi_{PJ}(X) = \mathrm{pf}\begin{bmatrix} \phi(\mathbf{g}_{11}) & \phi(\mathbf{g}_{12}) & \cdots & \phi(\mathbf{g}_{1N}) & \varphi(\mathbf{x}_1) \\ \phi(\mathbf{g}_{21}) & \phi(\mathbf{g}_{22}) & \cdots & \phi(\mathbf{g}_{2N}) & \varphi(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi(\mathbf{g}_{N1}) & \phi(\mathbf{g}_{N2}) & \cdots & \phi(\mathbf{g}_{NN}) & \varphi(\mathbf{x}_N) \end{bmatrix}$$

- Replace $\mathbf{x}_i$ with $(\mathbf{x}_i, \mathbf{h}_i^{(T)})$


0.430
0.425
0.420
0.415
0.410
0.405
0.400
E/E_{FG}
14
16
18
20
22
24
26
N
DMC-BCS
PJ-BF
Error

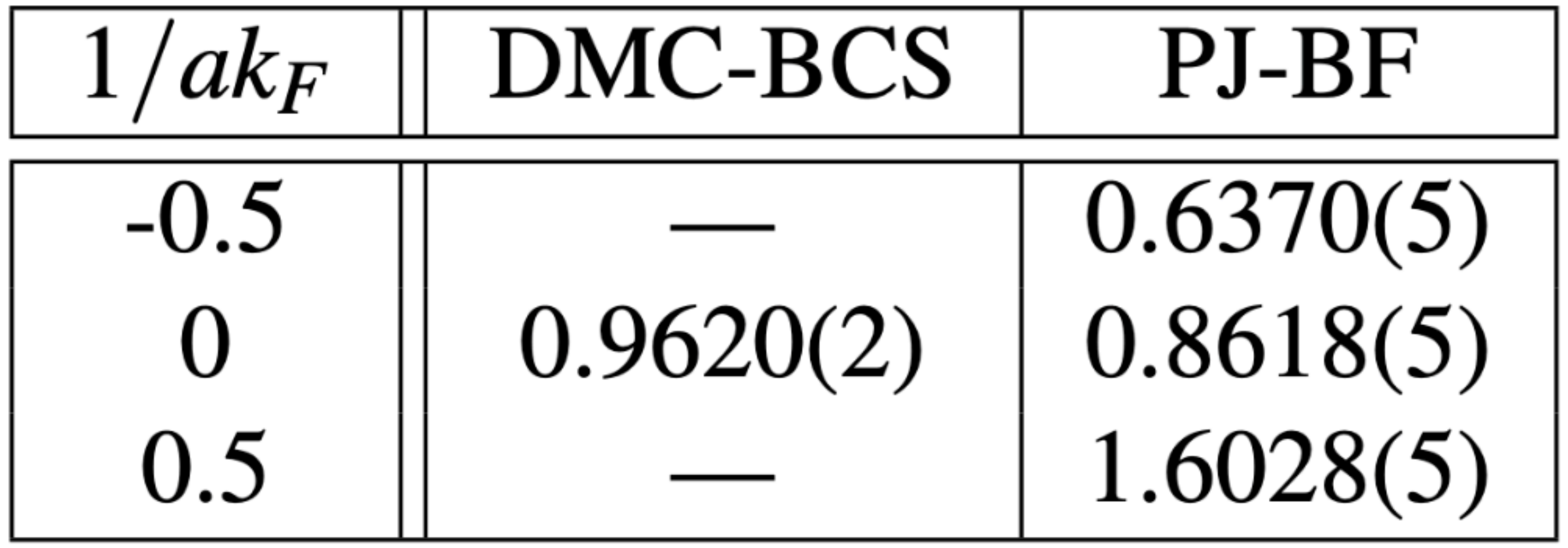
Pairing Gap

- At unitarity ($k_F r_e = 0.2$):



$\Delta(15)/E_{FG}$	DMC-BCS	PJ-BF
Non-translation invariant	0.9620(2)	0.8618(5)
Translation invariant	1.6475(2)	1.5263(5)

- Around unitarity ($k_F r_e = 0.2$):



$1/ak_F$	DMC-BCS	PJ-BF
-0.5	—	0.6370(5)
0	0.9620(2)	0.8618(5)
0.5	—	1.6028(5)

Conclusions and Perspectives

- Our Pfaffian ansatz is very general — works for any Hamiltonian (even those that exchange spin!)
- Can obtain lower energies than state-of-the-art diffusion Monte Carlo methods
- Requires far fewer parameters compared to other NQS applied to similar problems (~8500 vs millions)
- Future work:
  - Larger $N > 26$ and smaller r_e
  - Partially-polarized systems
  - Test if Jastrow is necessary

Thank you!

Deep Sets

- Introduced by Zaheer et al. in 2017 (arXiv:1703.06114)

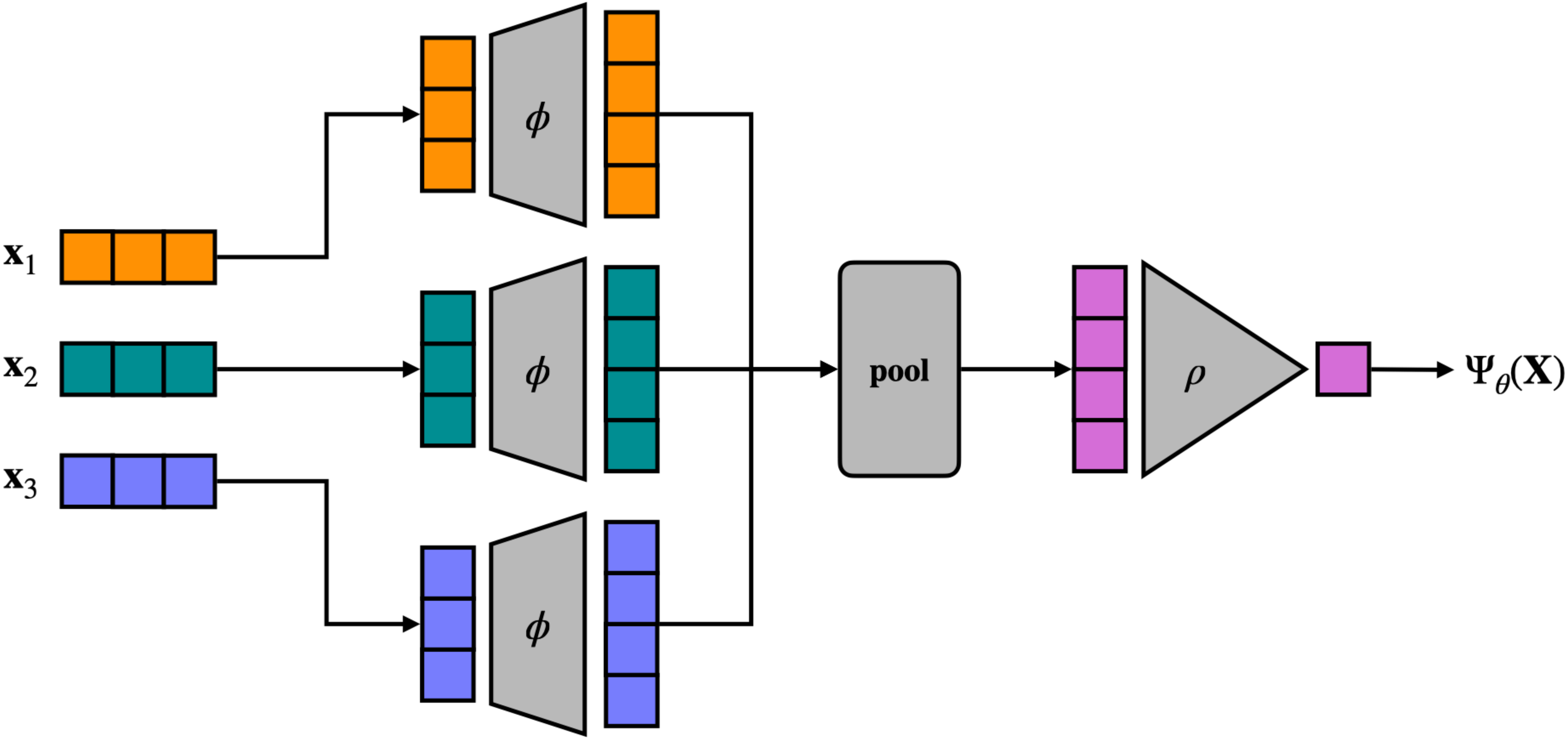
- Suppose we want to learn on sets of N elements $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$

- Any permutation-invariant function can be decomposed as

$$f(\{\mathbf{v}_i\}) = \rho\left(\mathrm{Pool}(\{\phi(\mathbf{v}_i)\})\right)$$

Pooling operation destroys ordering

- If ρ and ϕ are feedforward neural networks, then $f(\{\mathbf{v}_i\})$ is a universal approximator for permutation-invariant functions


$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
ϕ
ϕ
ϕ
pool
ρ
$\Psi_\theta(\mathbf{X})$