VARIATIONAL METHODS WITH NEURAL NETWORKS

JANE KIM

Ohio University

STREAMLINE Symposium

10 May 2024

STREAMLINE

SmarT Reduction and Emulation Applying Machine Learning In Nuclear Environments

QUANTUM MONTE CARLO

- Family of quantum many-body methods that use Monte Carlo integration to approximate high-dimensional integrals

- Variational Monte Carlo is the simplest method in the family

  - Construct a trial wave function $|\Psi_\theta\rangle$ with variational parameters θ

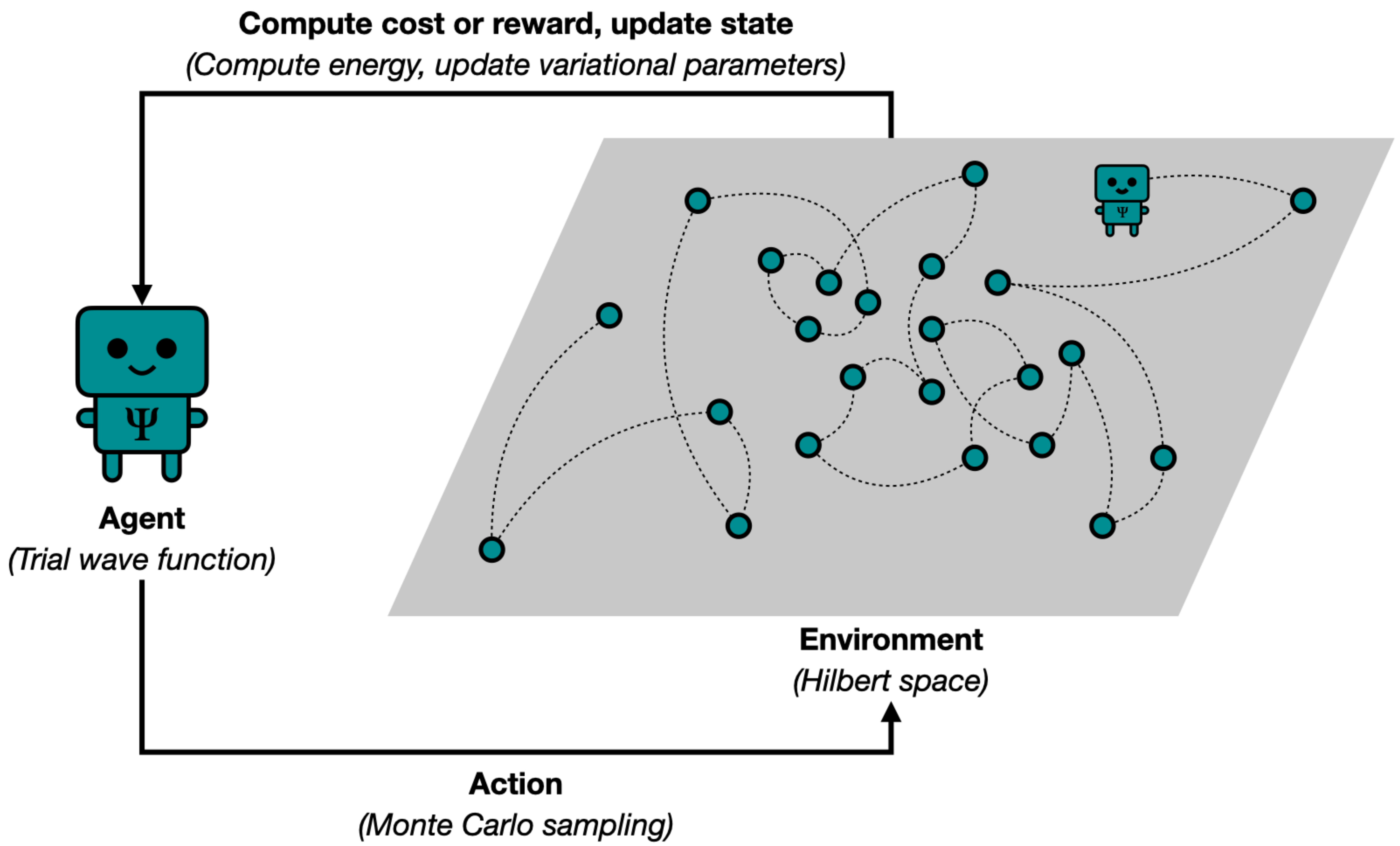
  - Optimize $|\Psi_\theta\rangle$ by minimizing $\langle E_\theta \rangle = \dfrac{\langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle}{\langle \Psi_\theta | \Psi_\theta \rangle}$

  - Usually used as a first step before more sophisticated methods, e.g. diffusion Monte Carlo (DMC), Green's function Monte Carlo, auxiliary-field quantum Monte Carlo, etc.

Sample $X \sim |\langle X | \Psi_\theta \rangle|^2$ → Compute $\langle E_\theta \rangle$ and $\nabla_\theta \langle E_\theta \rangle$ → Update parameters θ → (back to Sample)

NEURAL-NETWORK QUANTUM STATES (NQS)

- Use artificial neural networks to parameterize the trial wave function
- Train via variational Monte Carlo
- First used for spin systems by Carleo and Troyer in Science **355**, 602-606 (2017).
- Want a flexible NQS that has necessarily symmetries and boundary conditions
- Requires more parameters than traditional VMC



STOCHASTIC RECONFIGURATION

- Second-order optimization method based on imaginary-time propagation
- Closely related to the natural gradient method (Hessian of KL-divergence)
- Find $\Delta\theta$ such that

Replace with learning rate

$$|\Psi_{\theta+\Delta\theta}\rangle \approx e^{-\Delta\tau\hat{H}}|\Psi_\theta\rangle \implies S_\theta\Delta\theta = -\cancel{\frac{\Delta\tau}{2}}\nabla_\theta\langle E_\theta\rangle$$

- Quantum Fisher information matrix / geometric tensor

$$[S_\theta]_{ij} = \frac{\langle\partial_i\Psi_\theta|\partial_j\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} - \frac{\langle\partial_i\Psi_\theta|\Psi_\theta\rangle\langle\Psi_\theta|\partial_j\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}$$

- Improves training significantly
- Possible to use because our networks are reasonably small (~10000 parameters)

RECENT DEVELOPMENTS

communications physics

Article

https://doi.org/10.1038/s42005-024-01613-w

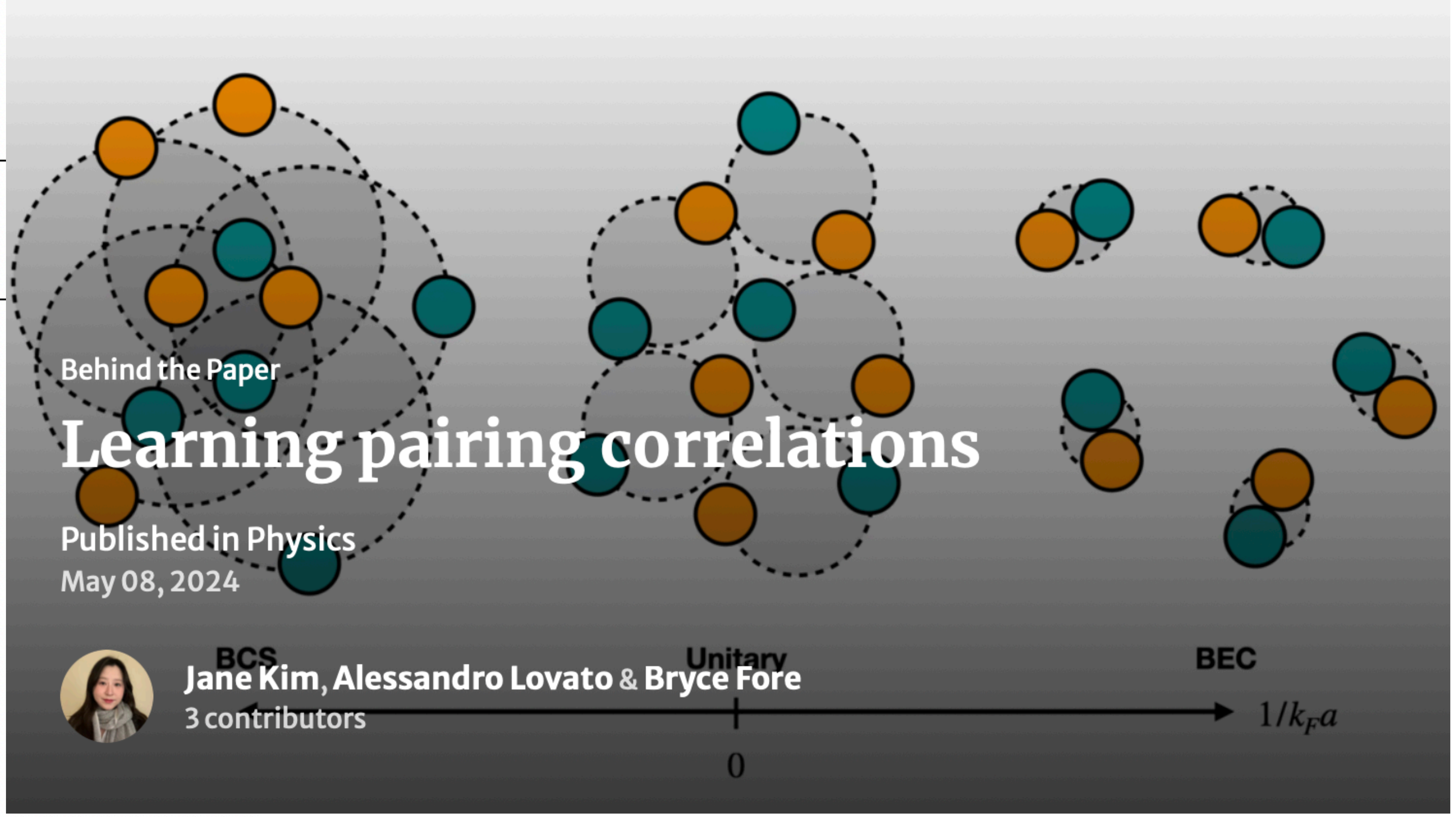
Neural-network quantum states for ultra-cold Fermi gases

Jane Kim[1,9], Gabriel Pescia[2,3], Bryce Fore[4], Jannes Nys[2,3], Giuseppe Carleo[2,3], Stefano Gandolfi[5], Morten Hjorth-Jensen[1,6] & Alessandro Lovato[4,7,8]

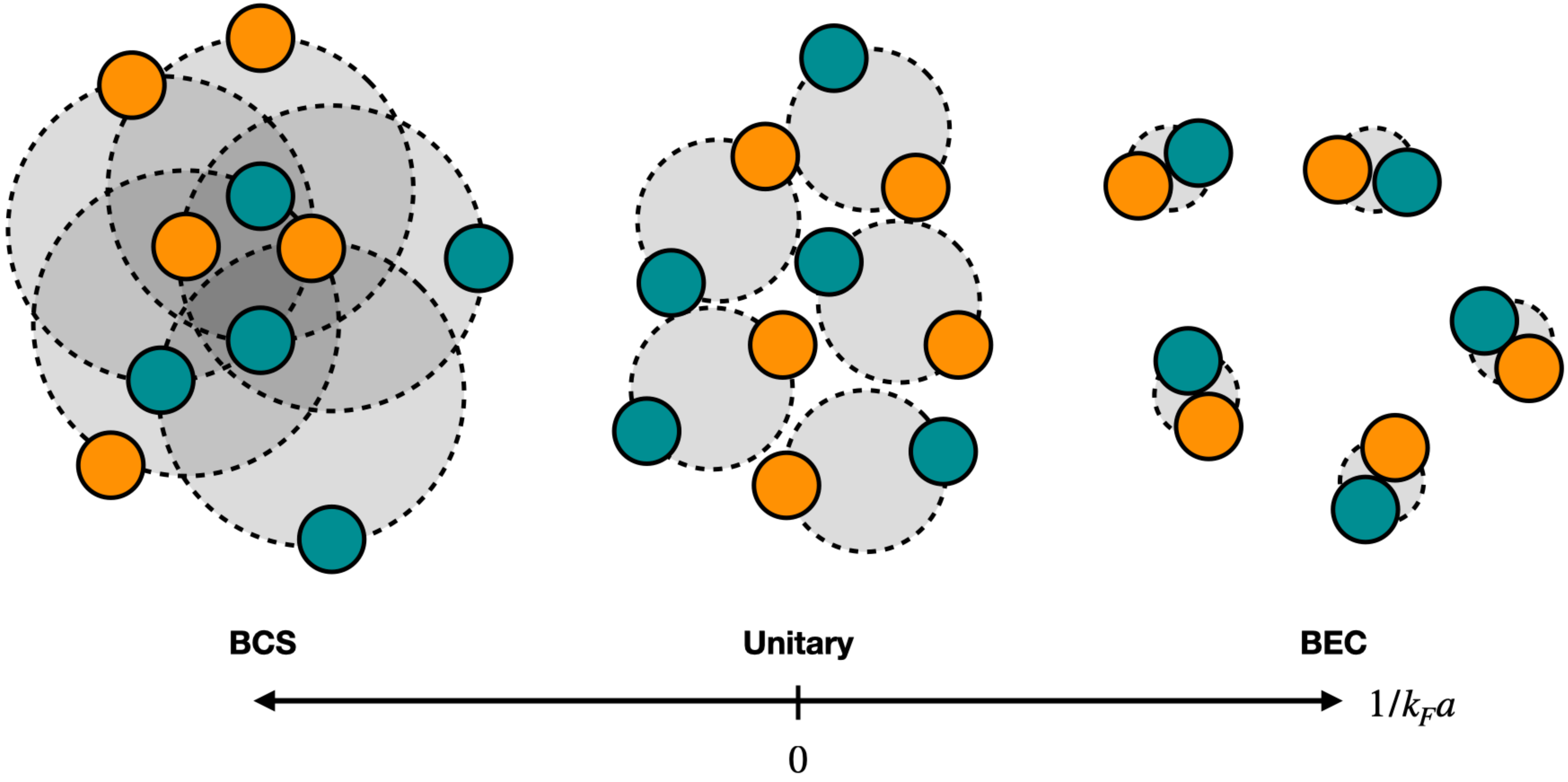
Ultra-cold Fermi gases exhibit a rich array of quantum mechanical properties, including the transition from a fermionic superfluid Bardeen-Cooper-Schrieffer (BCS) state to a bosonic superfluid Bose-Einstein condensate (BEC). While these properties can be precisely probed experimentally, accurately describing them poses significant theoretical challenges due to strong pairing correlations and the non-perturbative nature of particle interactions. In this work, we introduce a Pfaffian-Jastrow neural-network quantum state featuring a message-passing architecture to efficiently capture pairing and backflow correlations. We benchmark our approach on existing Slater-Jastrow frameworks and state-of-the-art diffusion Monte Carlo methods, demonstrating a performance advantage and the scalability of our scheme. We show that transfer learning stabilizes the training process in the presence of strong, short-ranged interactions, and allows for an effective exploration of the BCS-BEC crossover region. Our findings highlight the potential of neural-network quantum states as a promising strategy for investigating ultra-cold Fermi gases.

JK, MHJ, AL *et al. Commun. Phys.* 7, 148 (2024).

Behind the paper



THE BCS-BEC CROSSOVER



PFAFFIAN WAVE FUNCTION

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

$$\Phi_{PJ}(X) = \mathrm{pf}\begin{bmatrix} 0 & \phi(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}_N) \\ -\phi(\mathbf{x}_2, \mathbf{x}_1) & 0 & \cdots & \phi(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ -\phi(\mathbf{x}_N, \mathbf{x}_1) & -\phi(\mathbf{x}_N, \mathbf{x}_2) & \cdots & 0 \end{bmatrix}$$

- Pairing orbital into commonly decomposed into explicit singlet and triplet contributions

- 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

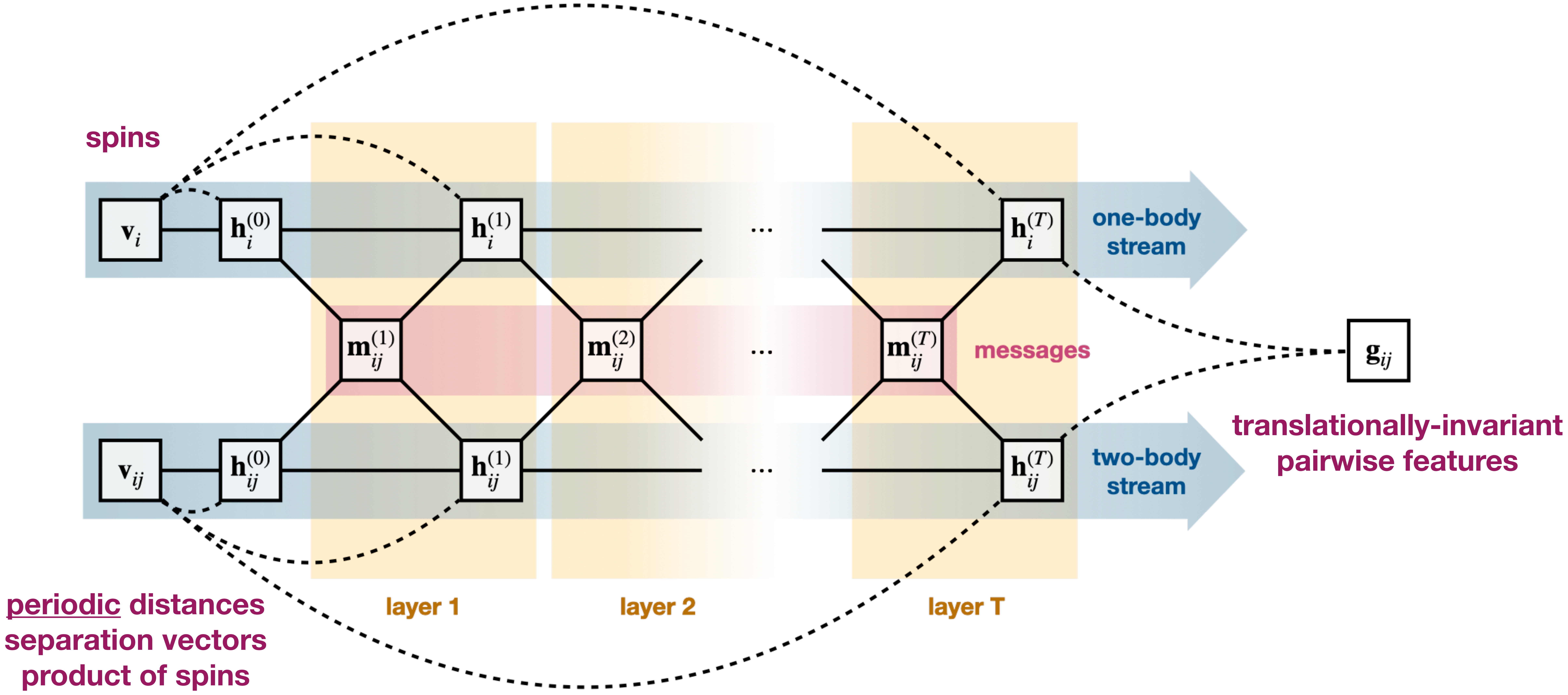
MESSAGE-PASSING NEURAL NETWORK

- Permutation-equivariant graph neural network
- Represent quantum system by a fully-connected graph
- Iteratively build backflow correlations into new one- and two-body features
- Skip connections help avoid vanishing gradient problem
- 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: $\mathbf{h}_i^{(0)} = (\mathbf{v}_i, A\mathbf{v}_i)$

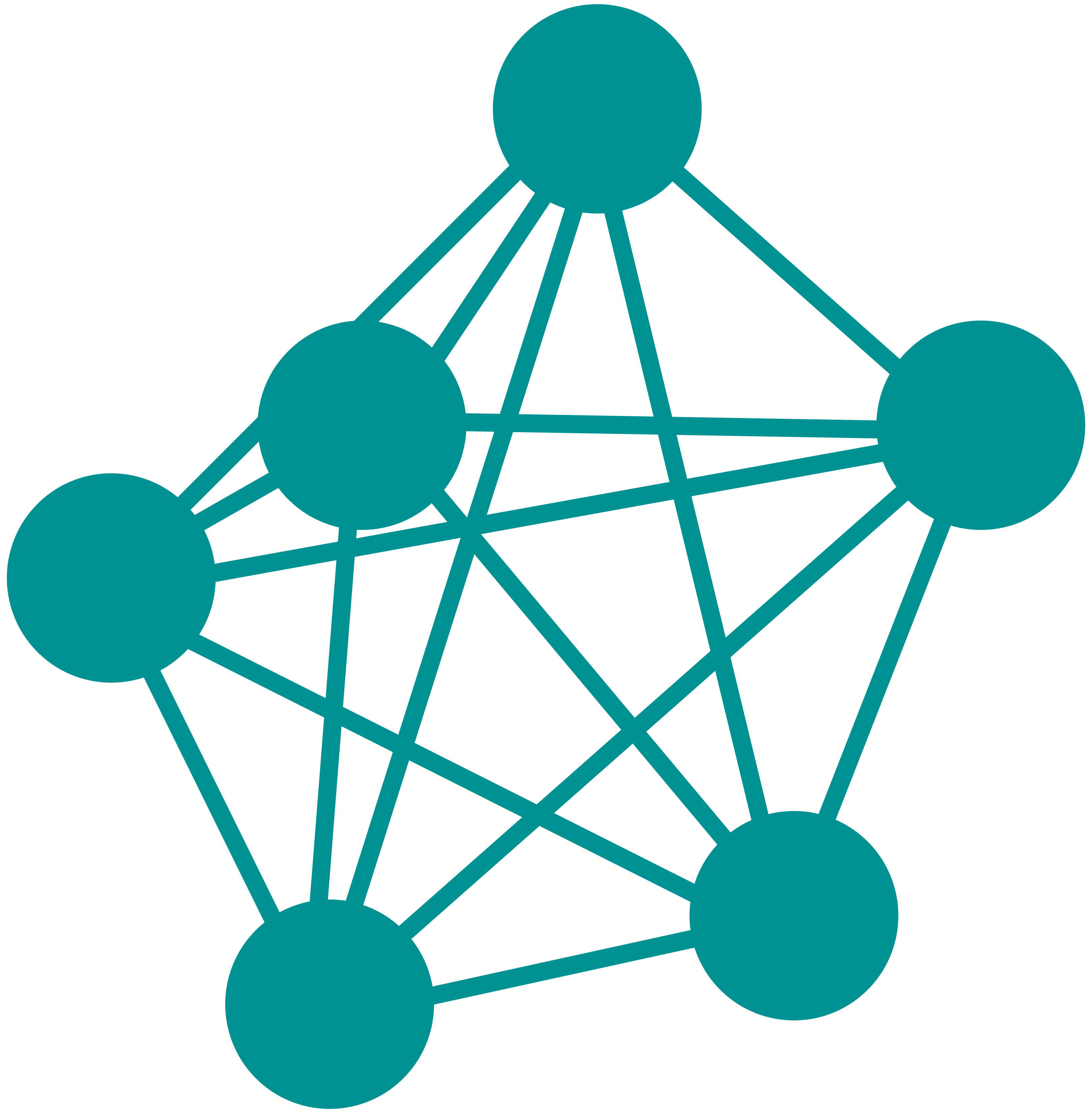
$$\mathbf{h}_{ij}^{(0)} = (\mathbf{v}_{ij}, B\mathbf{v}_{ij})$$

The set-up

MESSAGE-PASSING NEURAL NETWORK



MESSAGE-PASSING NEURAL NETWORK



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

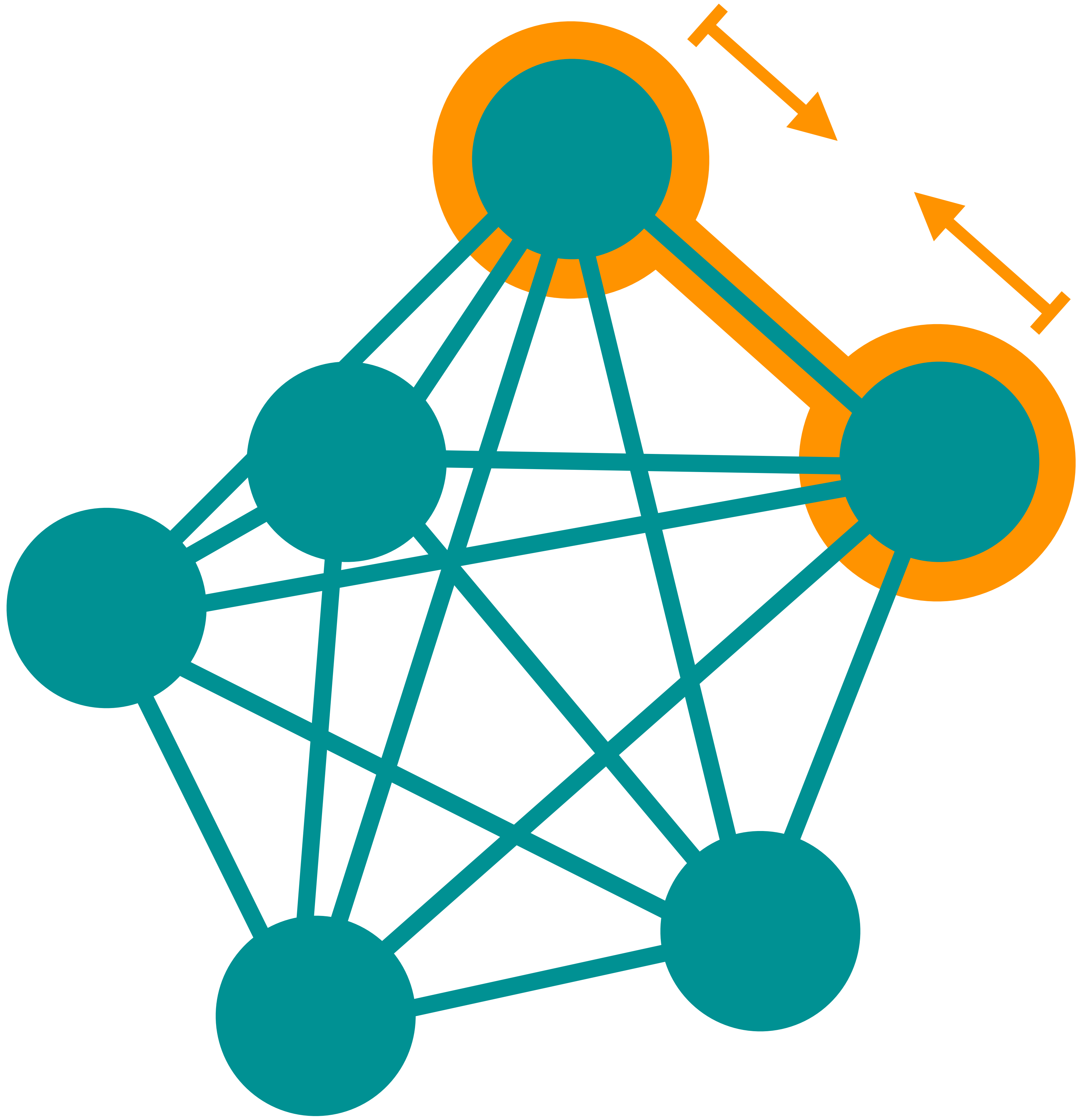
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

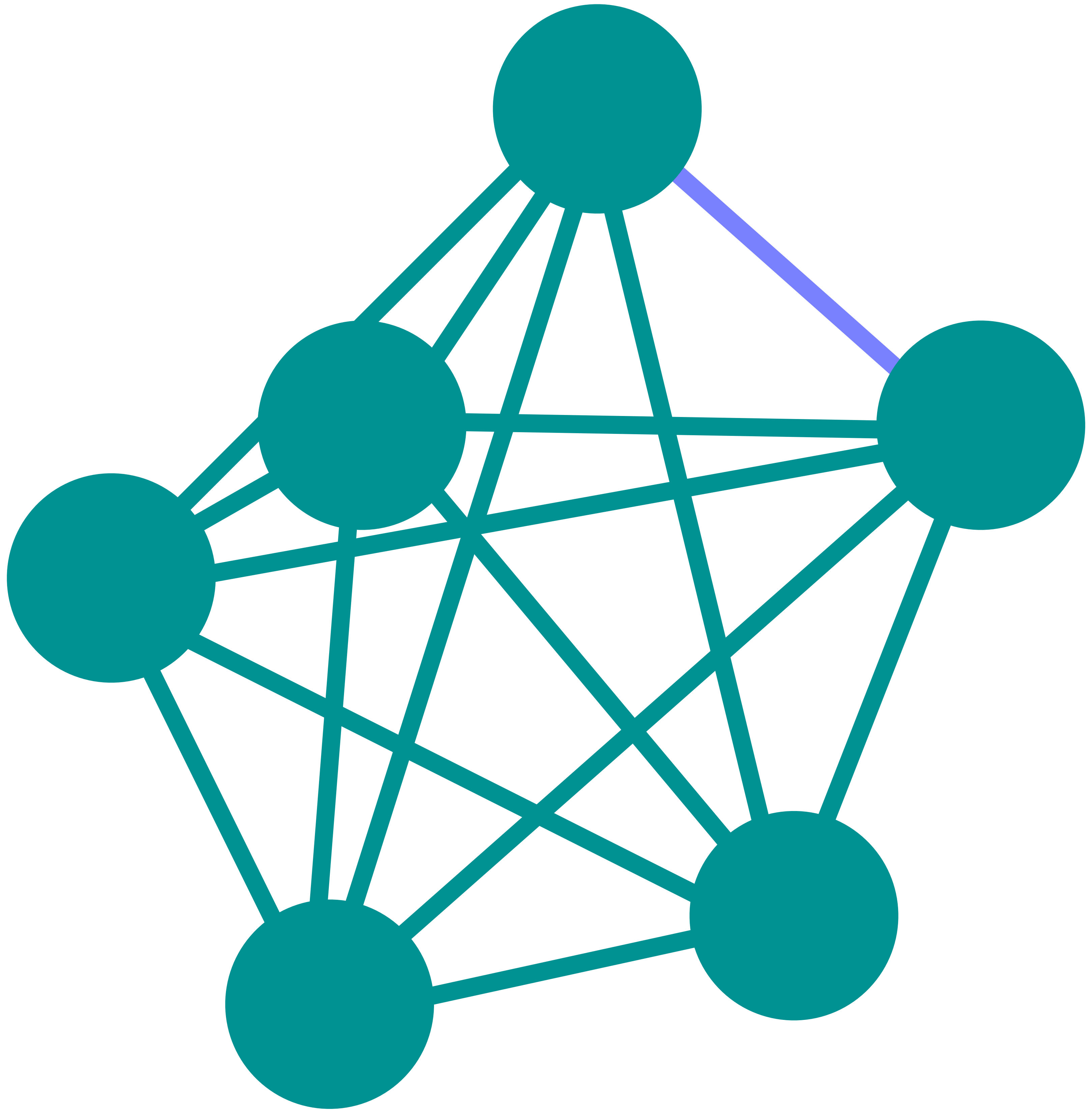
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

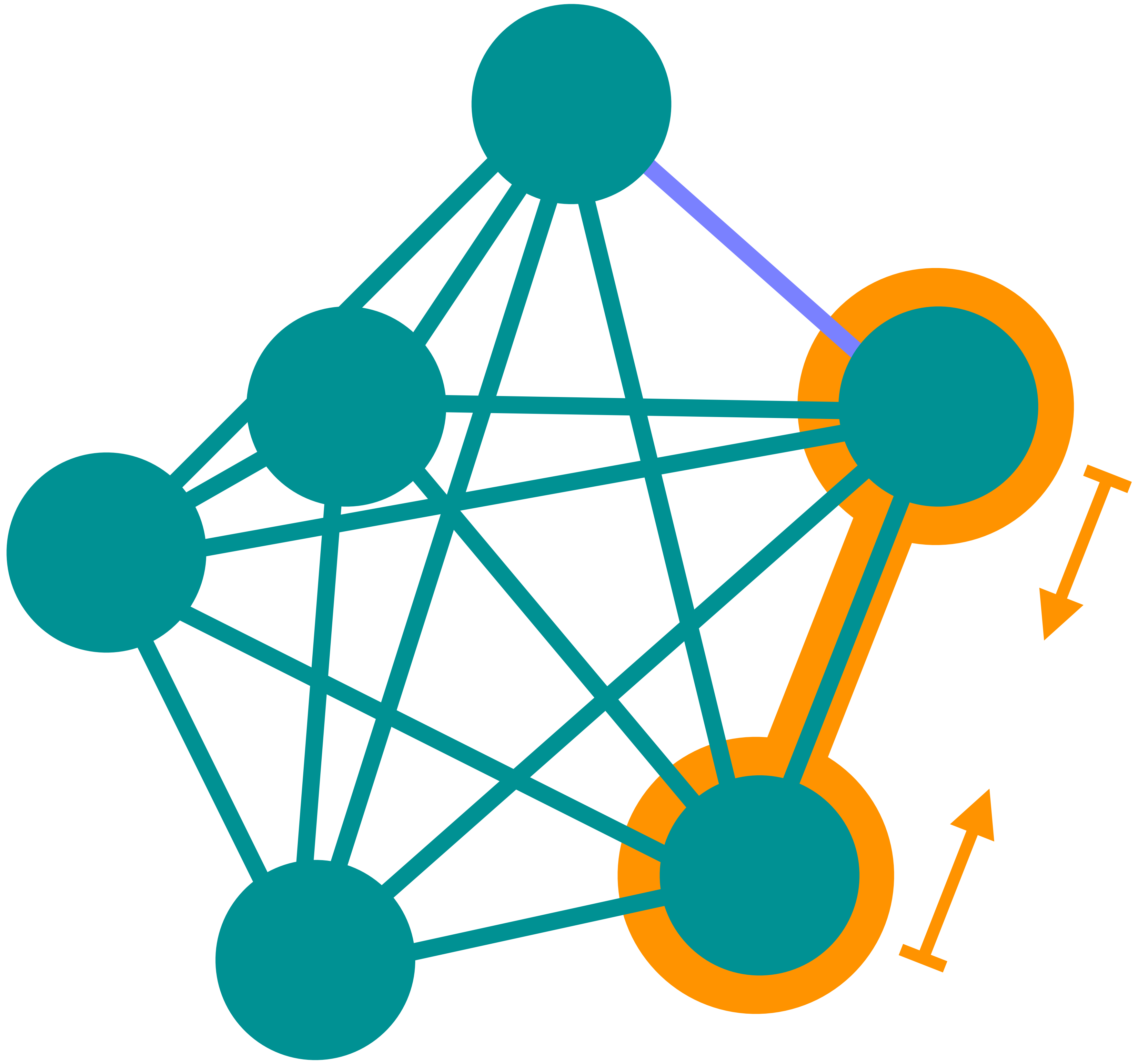
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

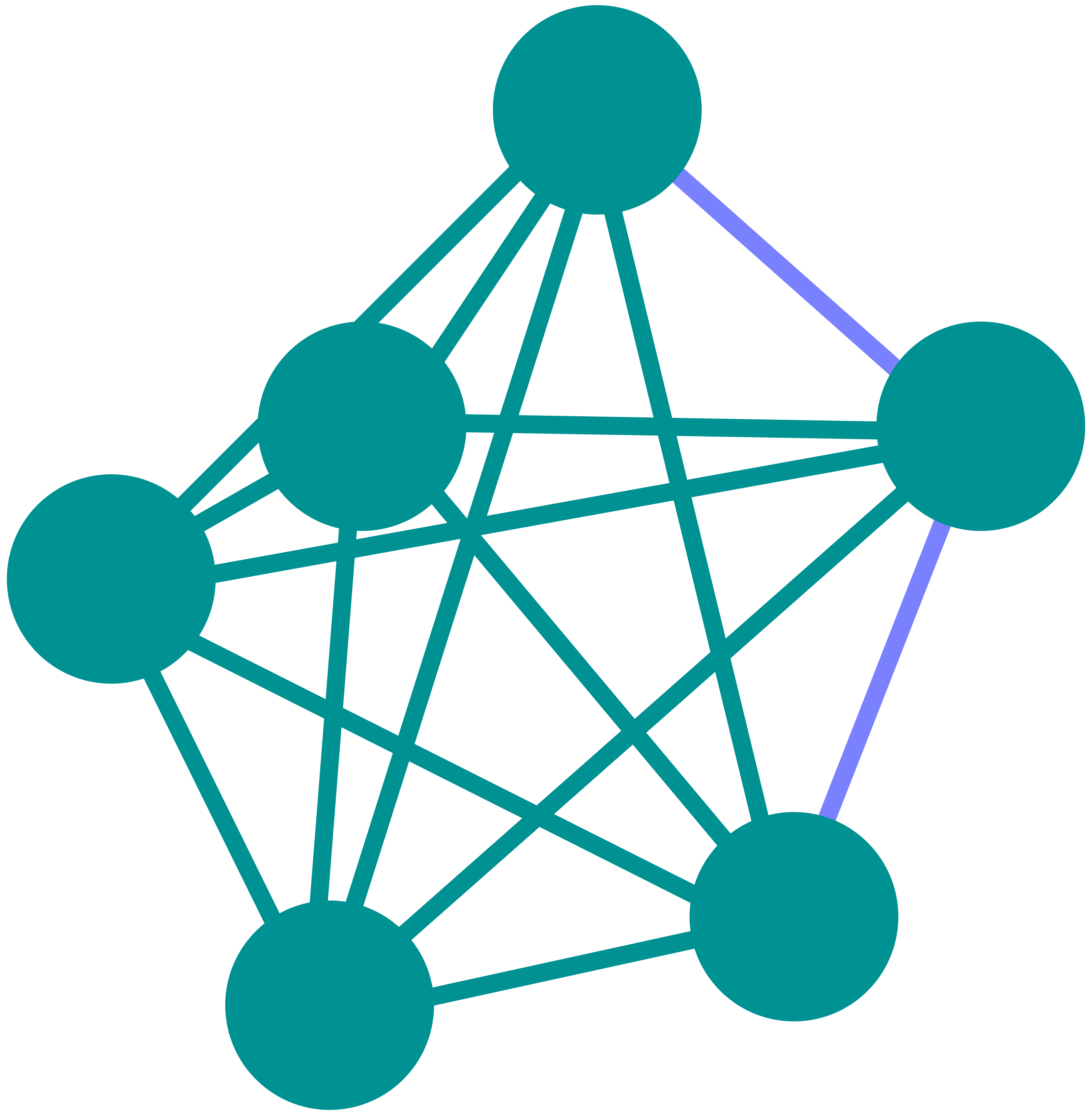
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

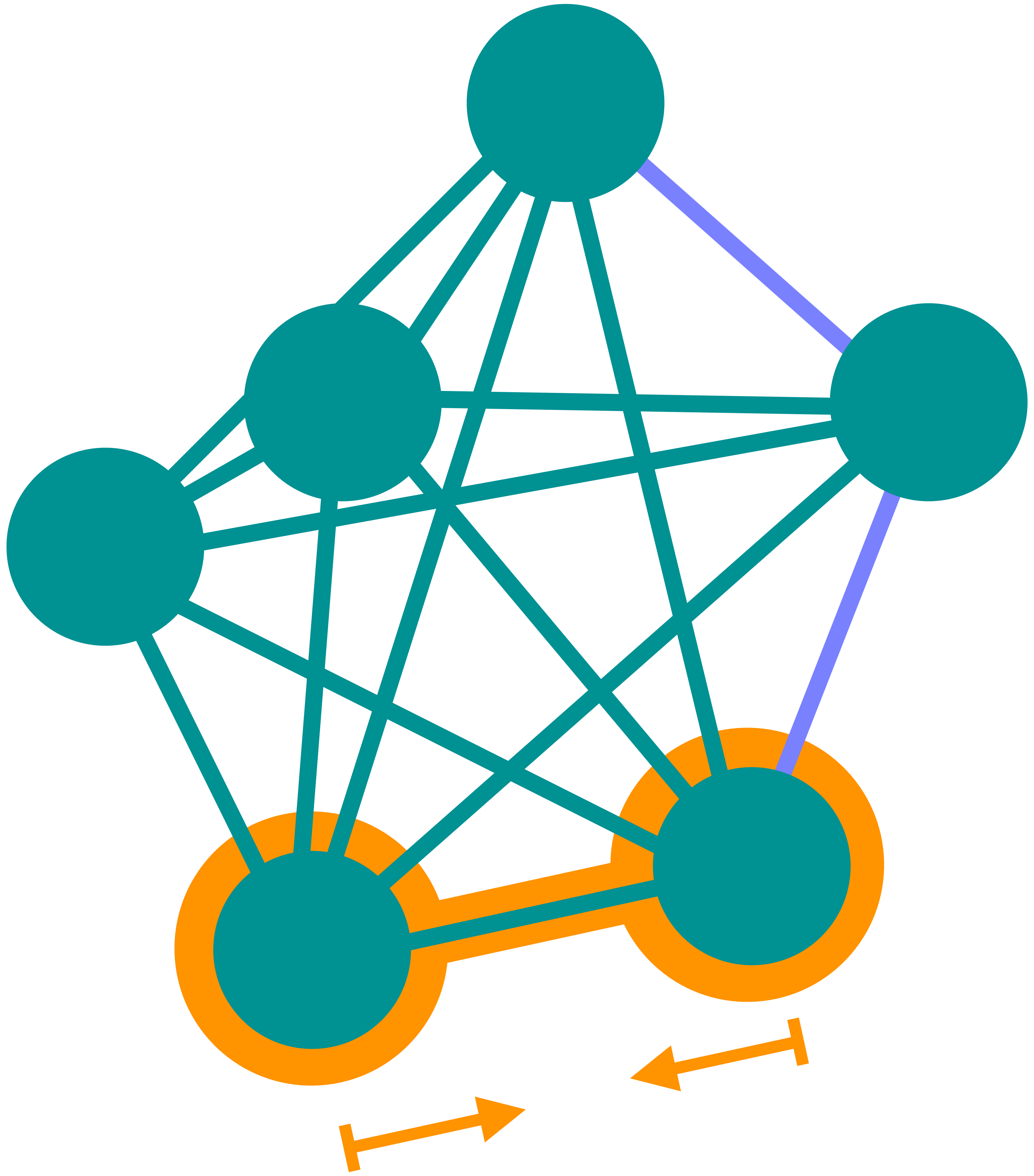
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

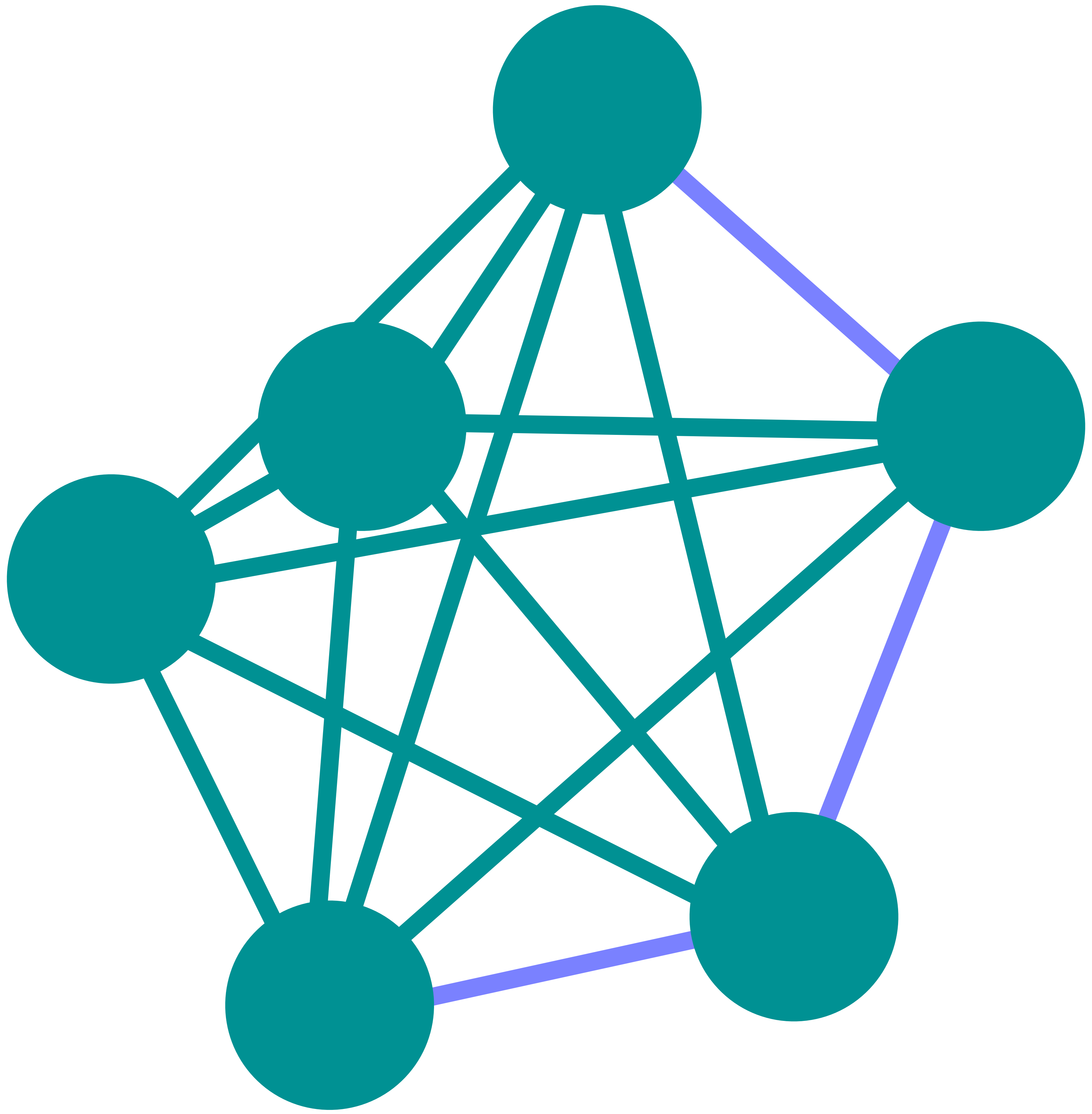
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

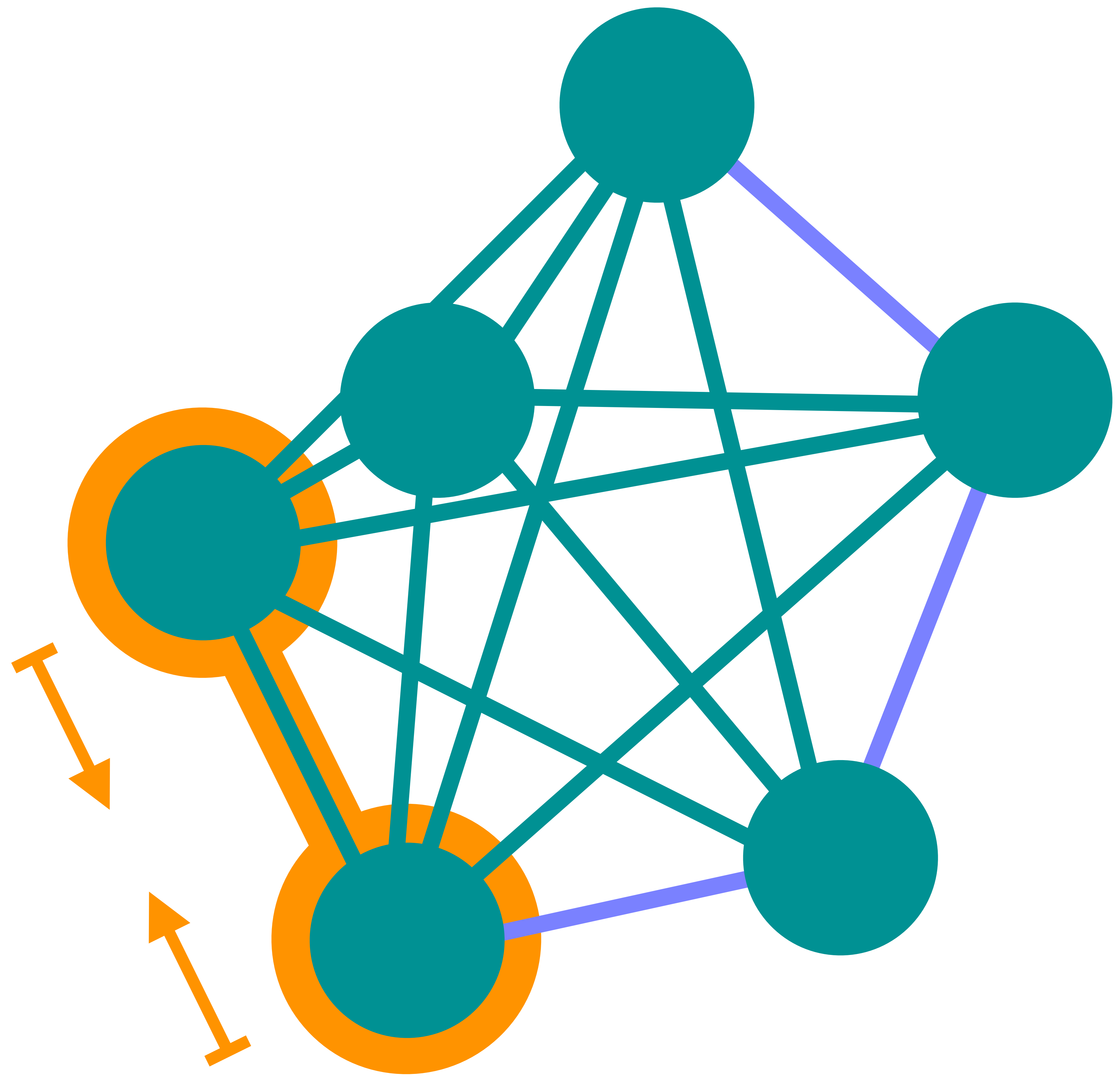
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

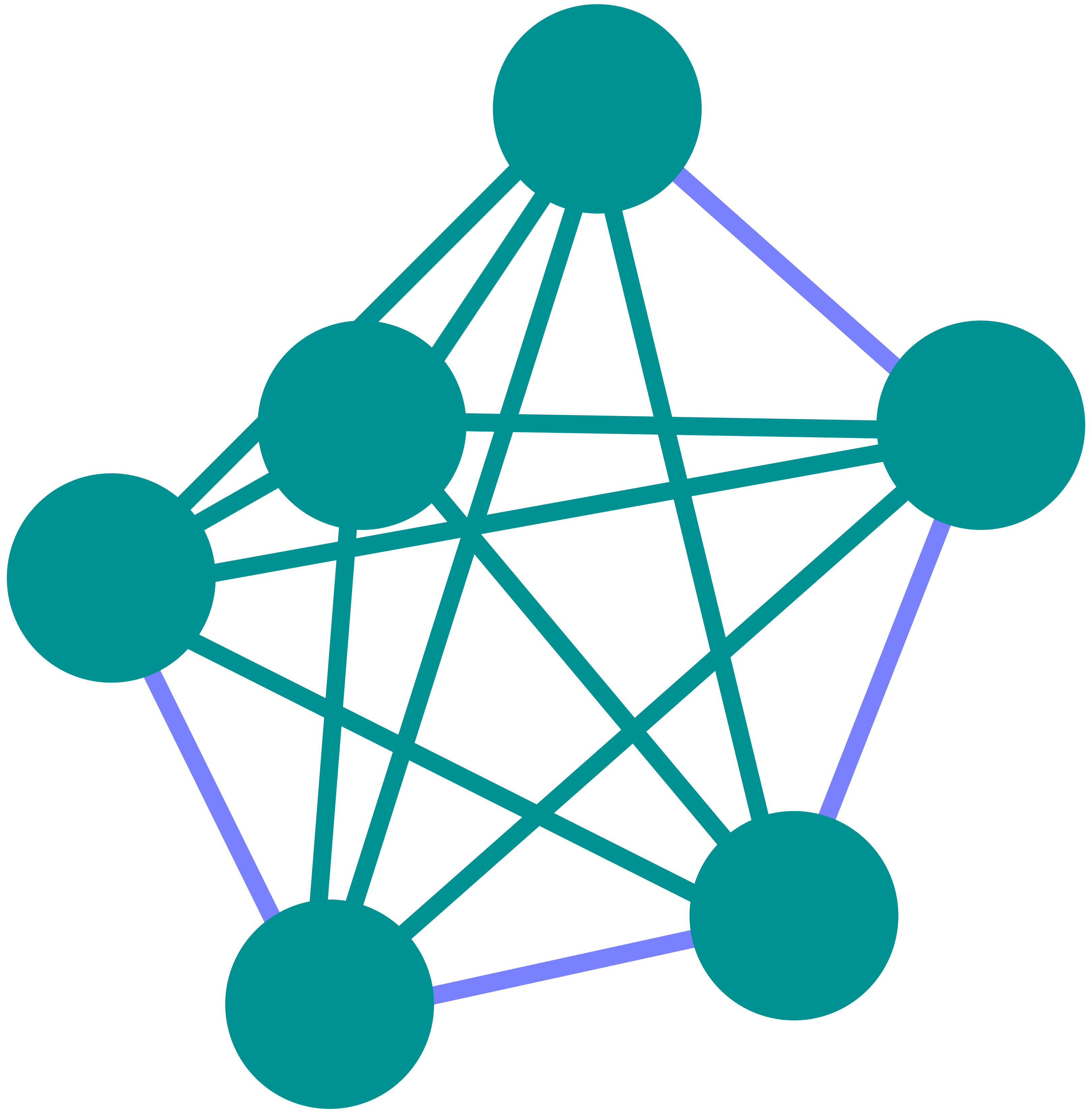
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

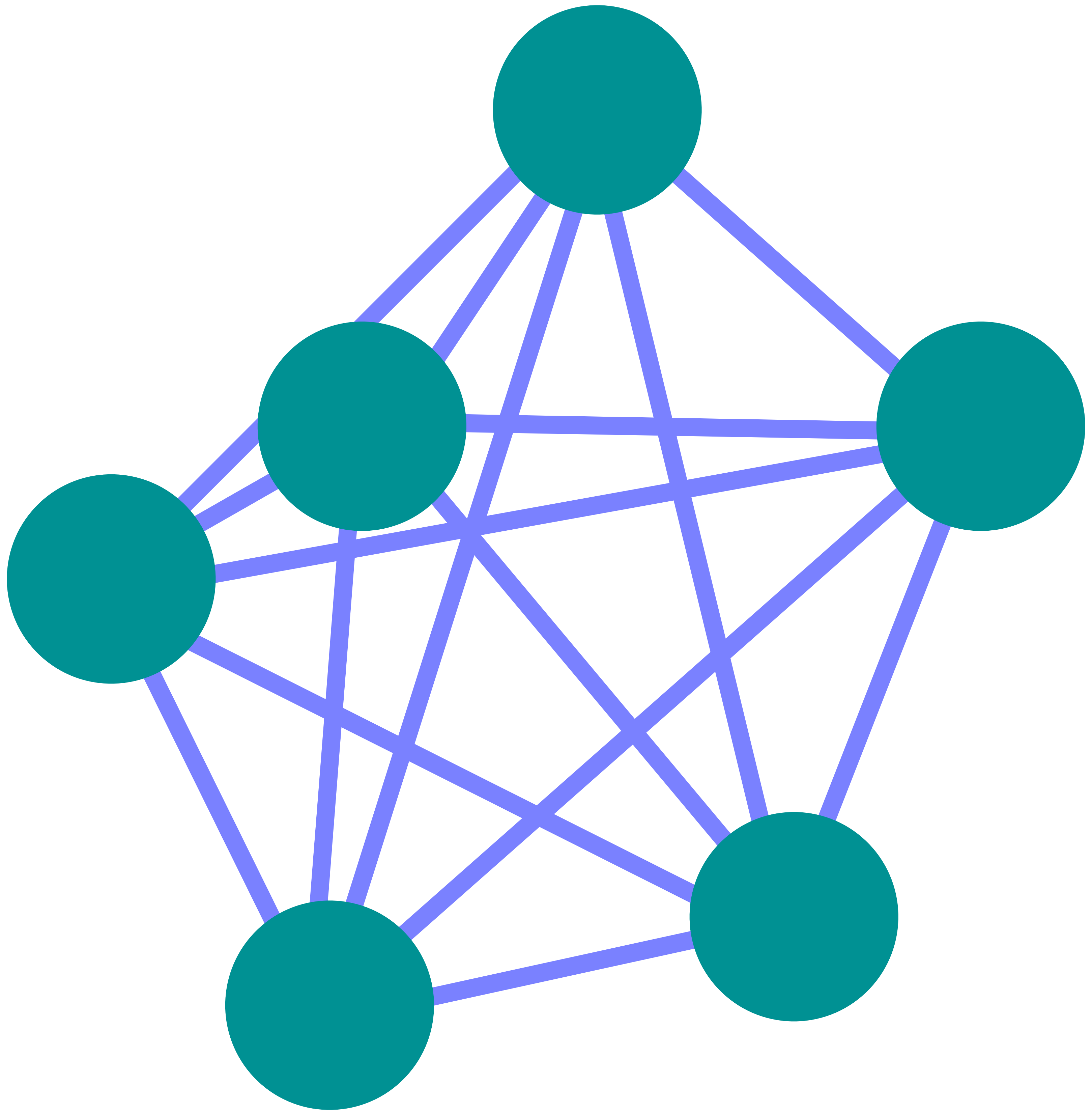
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

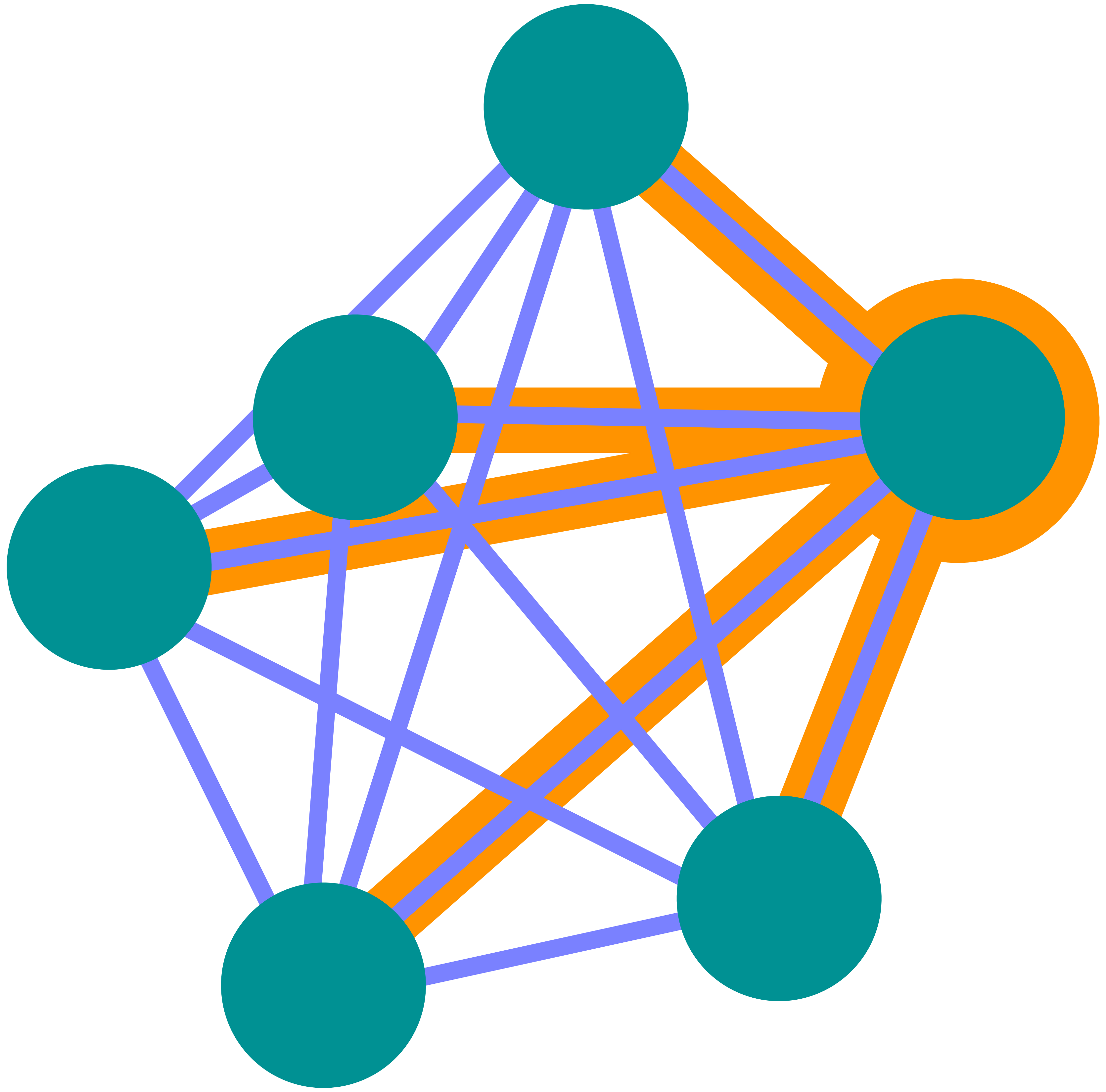
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

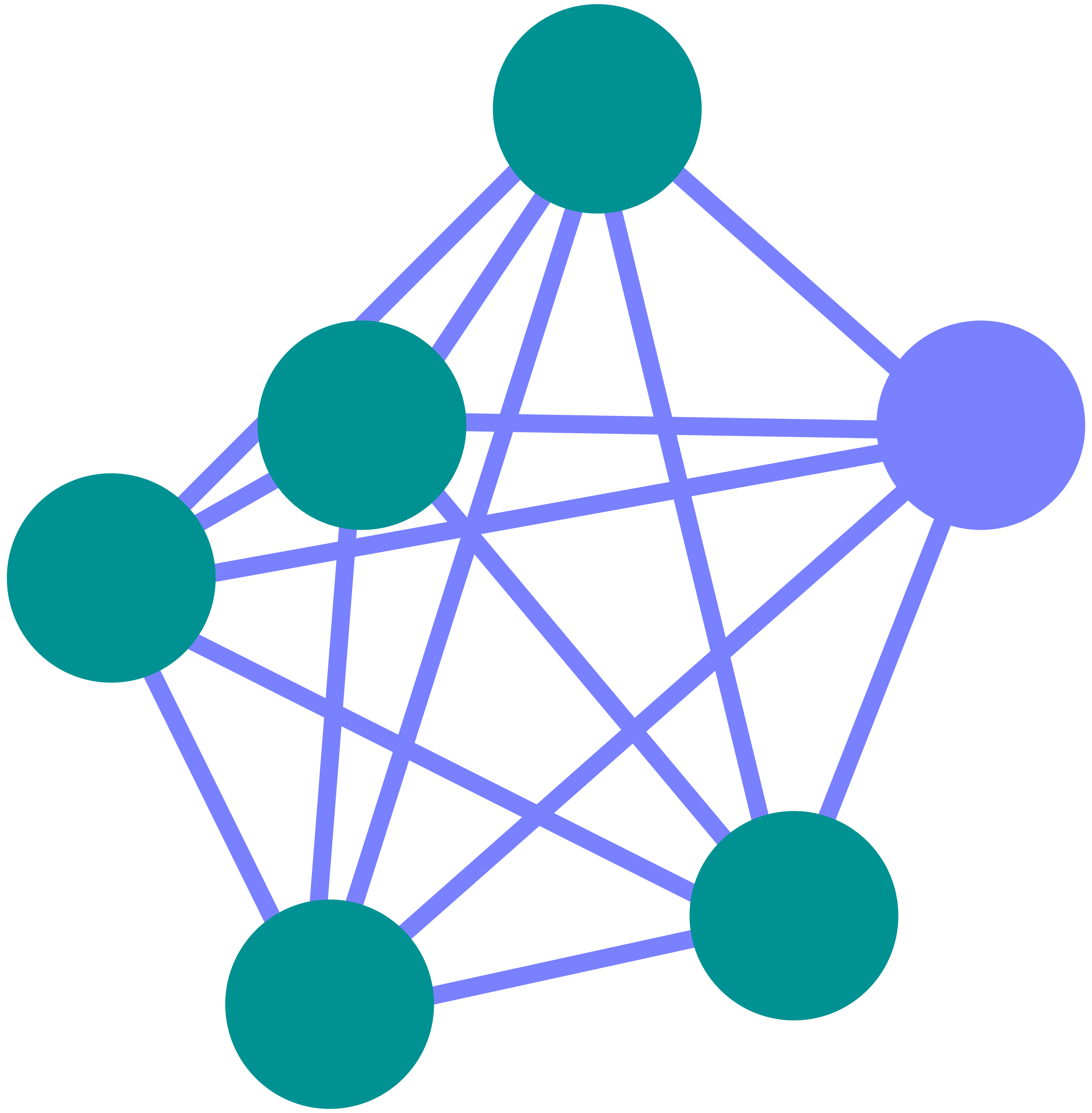
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

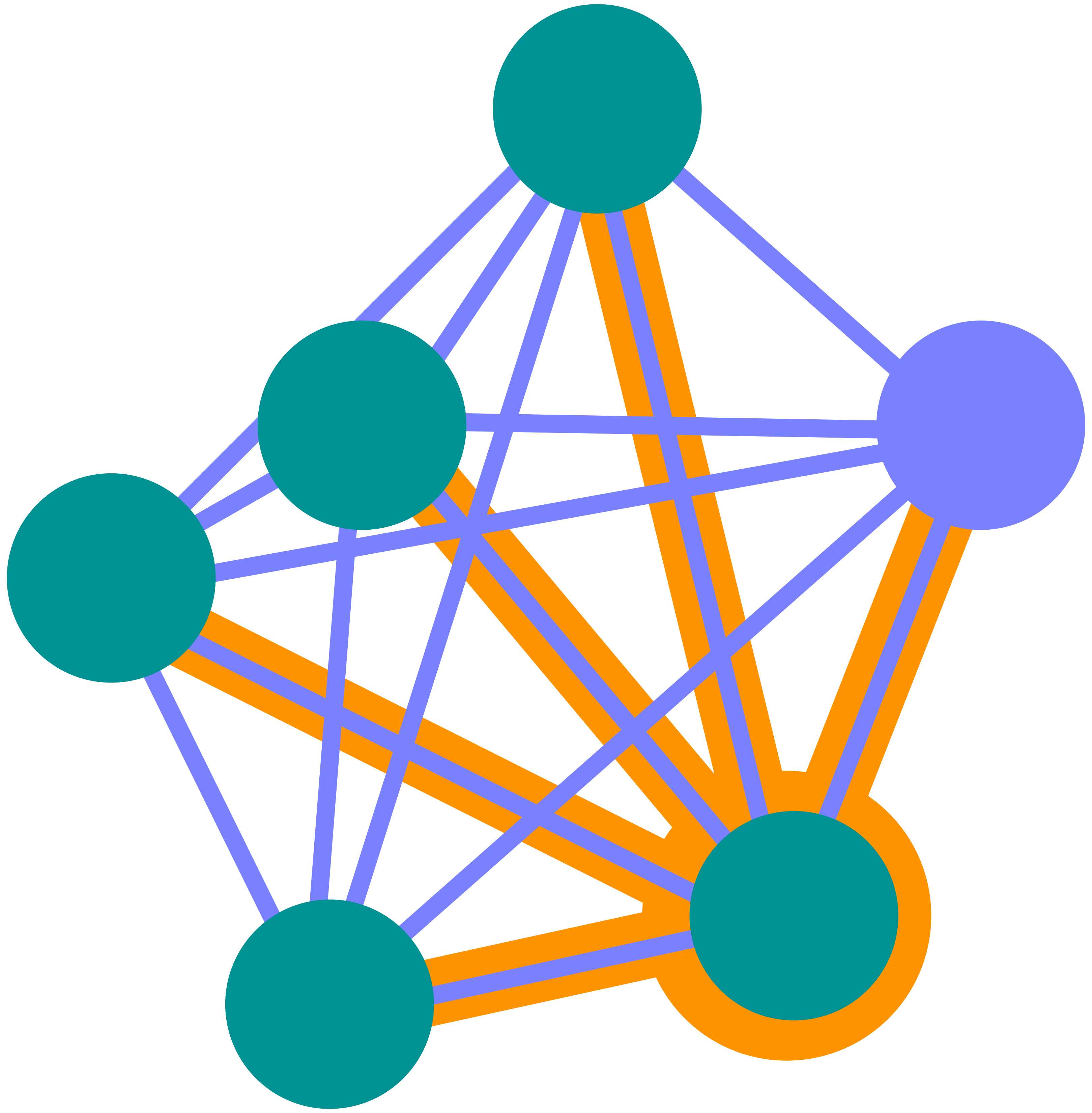
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

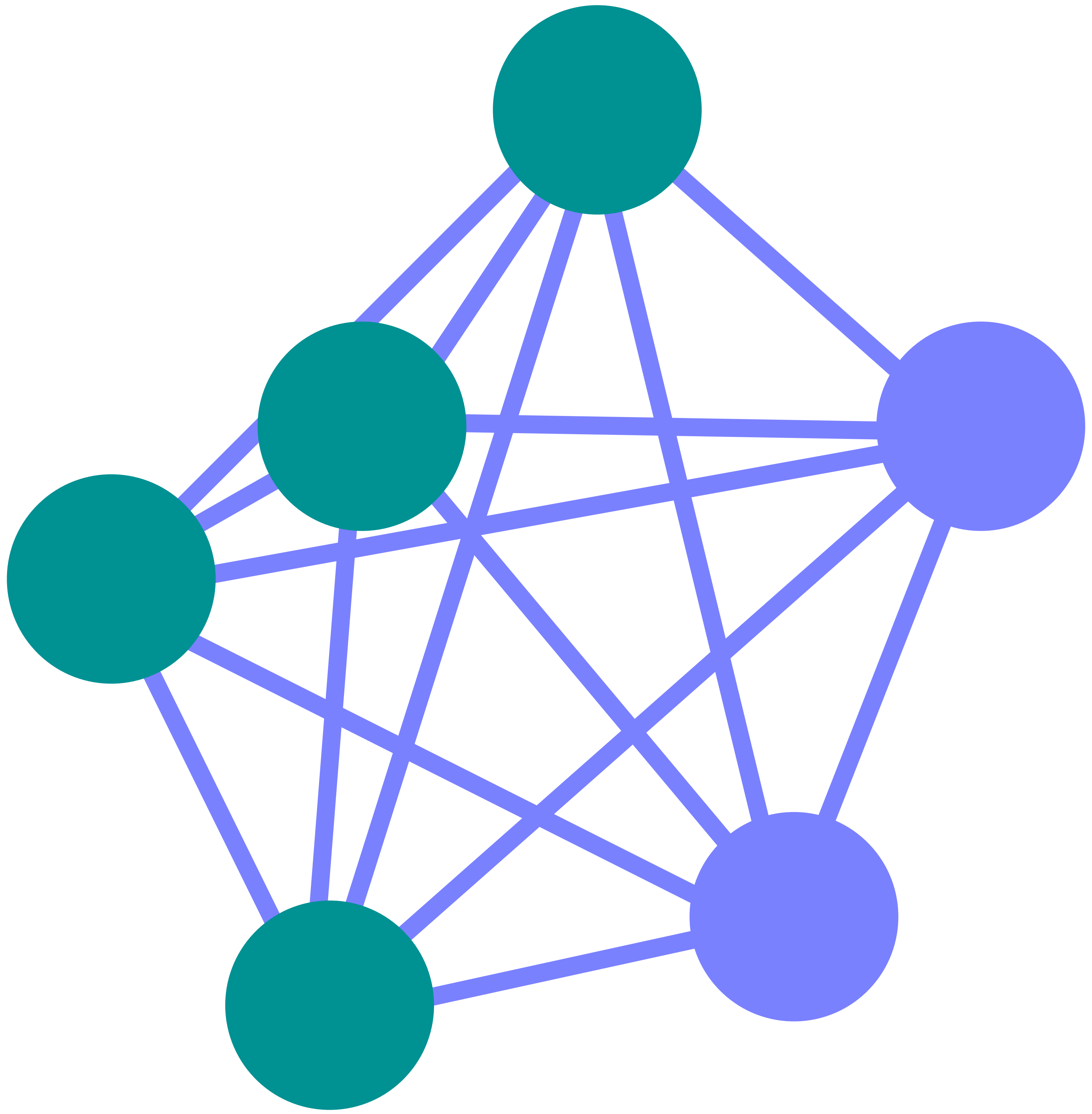
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

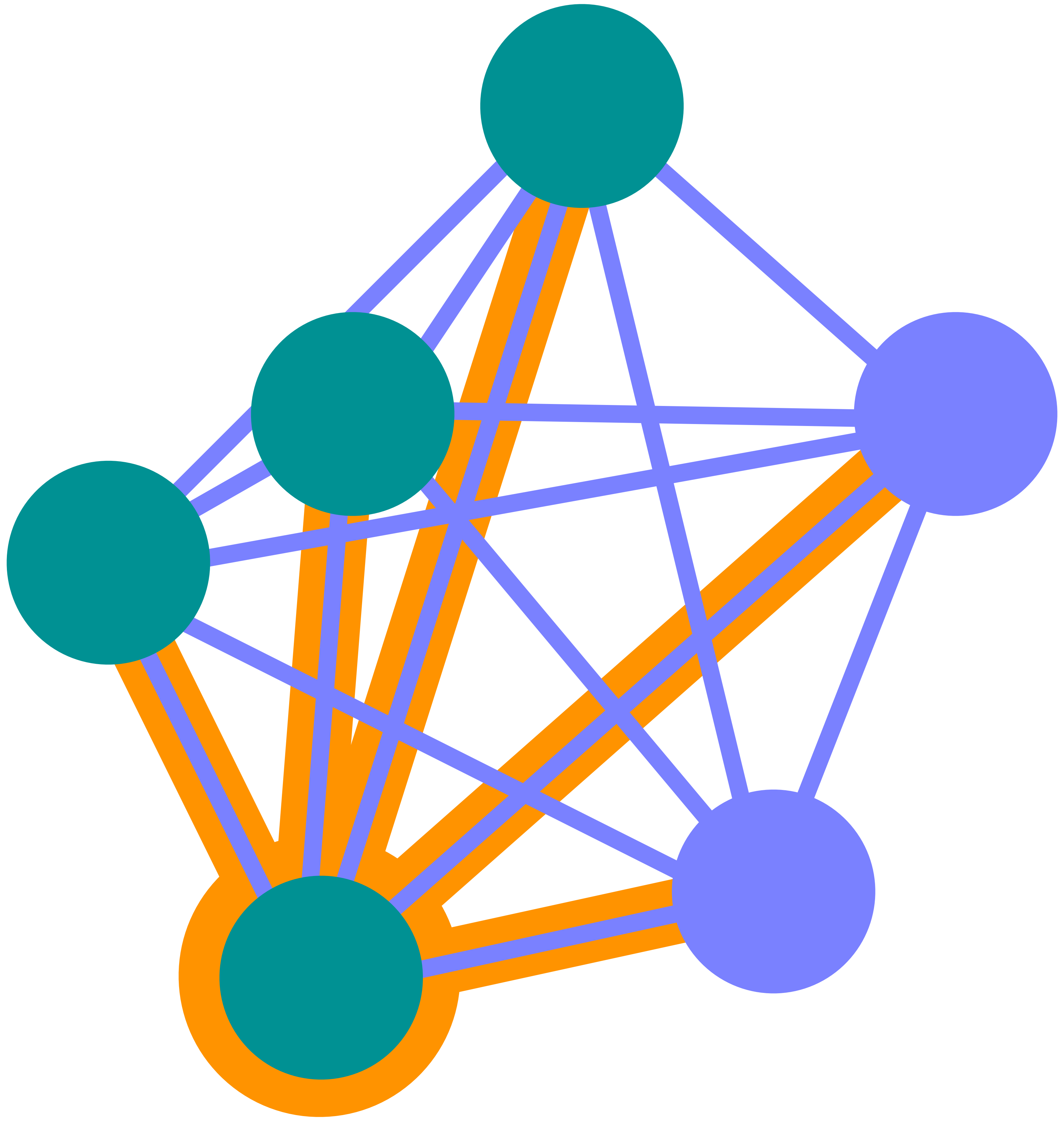
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

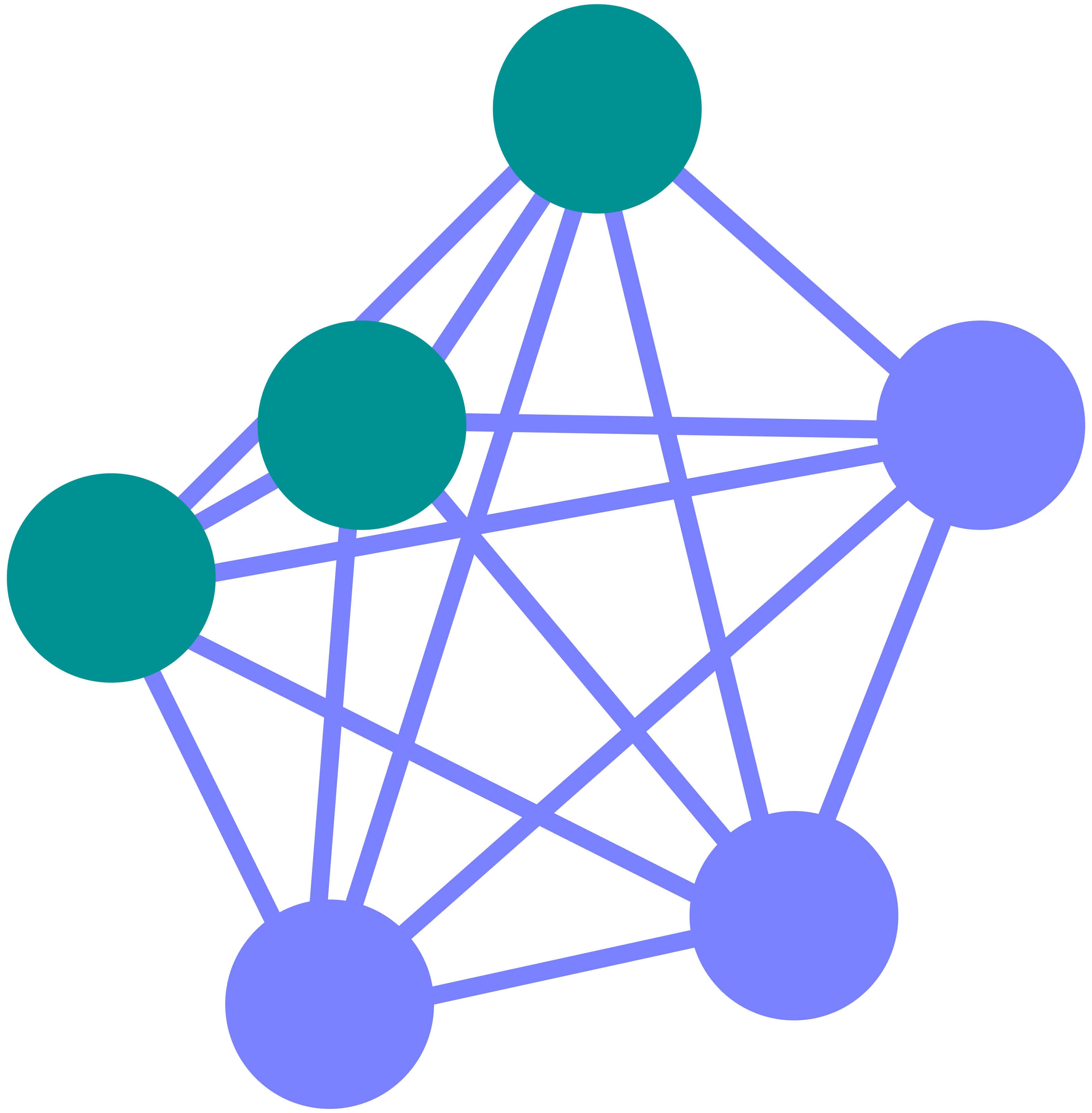
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

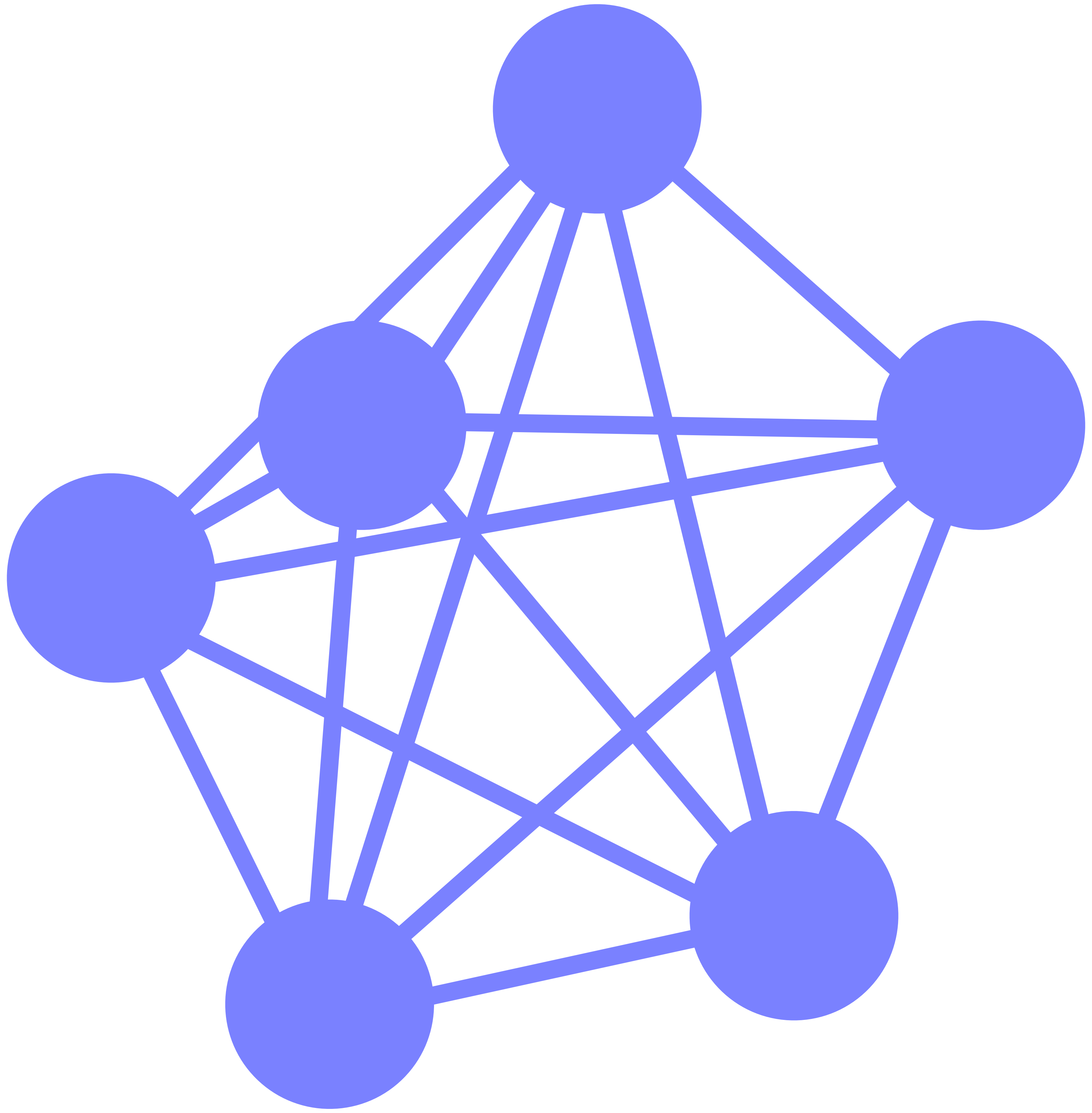
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

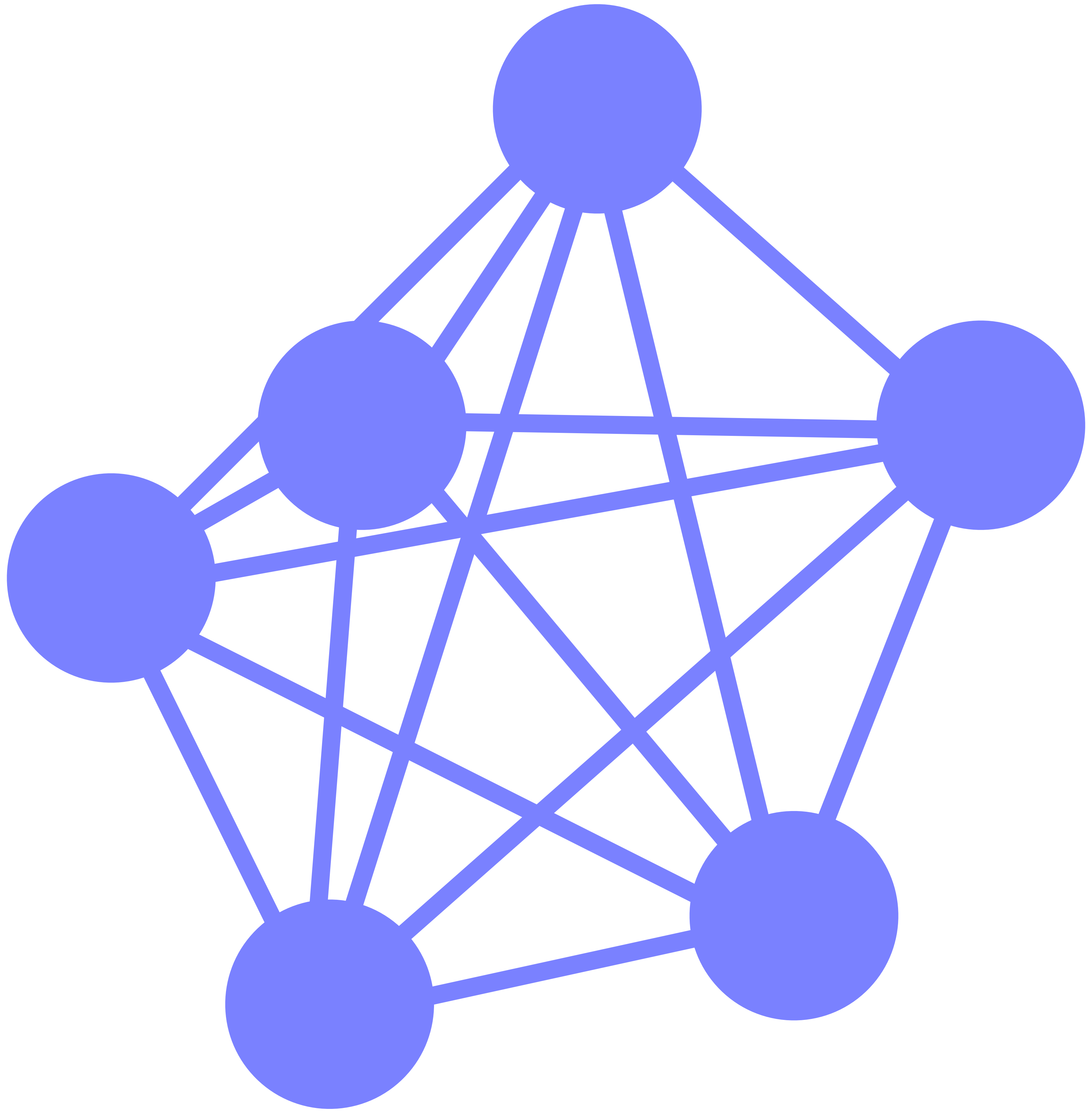
$$\mathbf{m}_{ij}^{(t)} = \mathbf{M}_t\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{h}_{ij}^{(t-1)}\right)$$

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

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

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

MESSAGE-PASSING NEURAL NETWORK



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

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

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

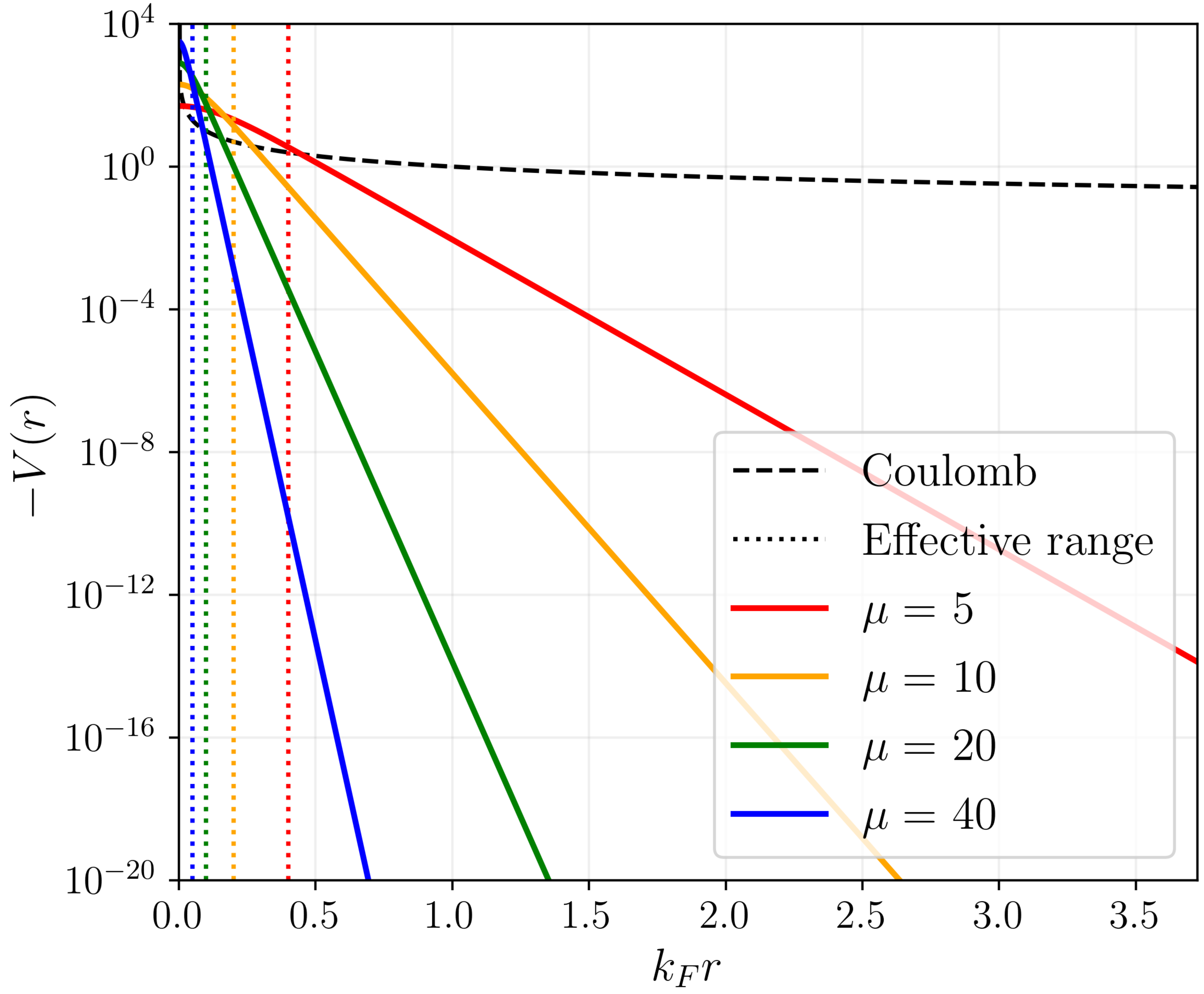
$$\mathbf{m}_i^{(t)} = \mathrm{Pool}\left(\{\mathbf{m}_{ij}^{(t)} | j \neq i\}\right)$$

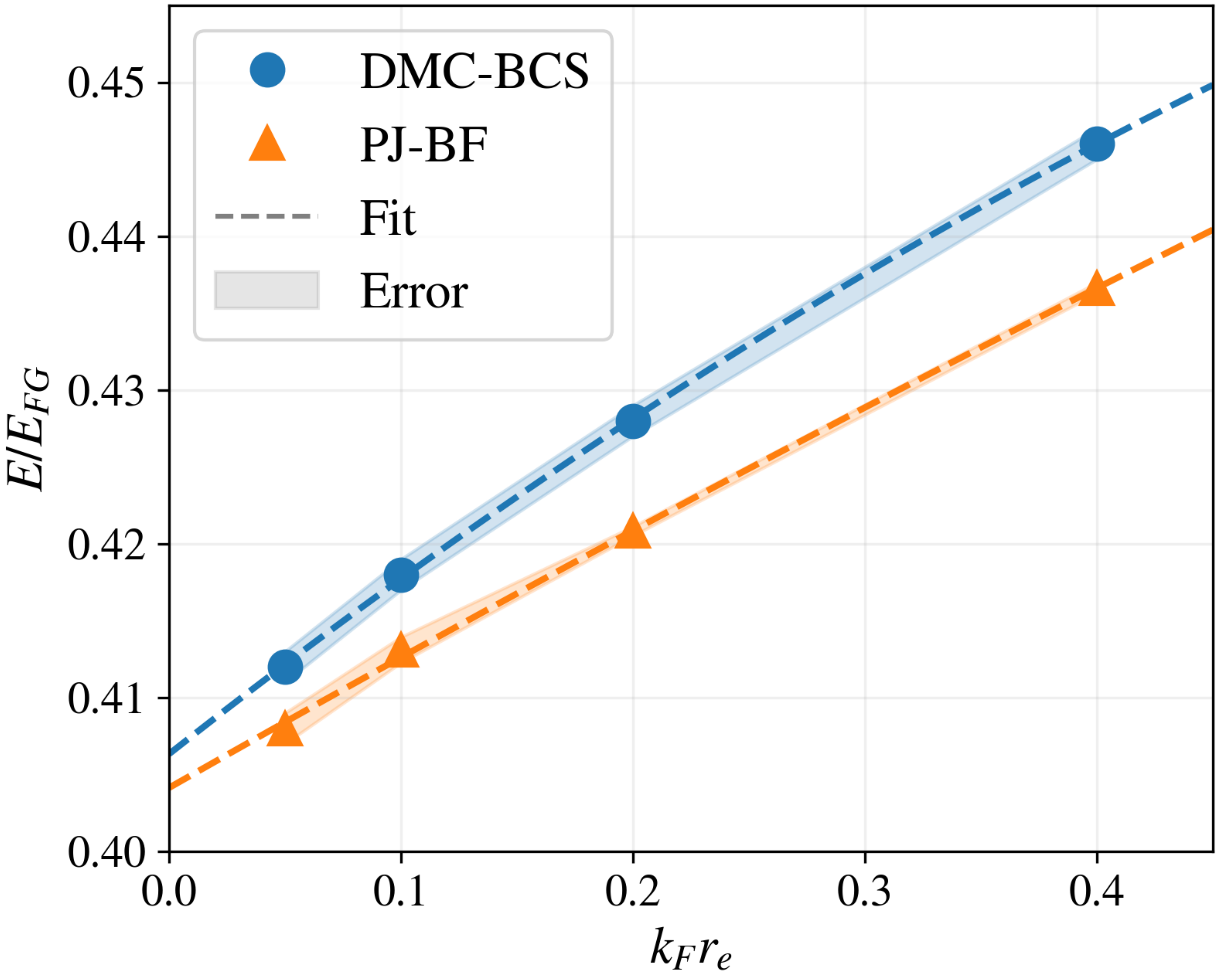
$$\mathbf{h}_i^{(t)} = \left(\mathbf{v}_i, \boxed{\mathbf{F}_t}\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)\right)$$

Feedforward
neural networks

TRANSFER LEARNING

- Pretraining a network on easier or related tasks





N-INDEPENDENCE

- The total number of parameters for even N and rectangular nets:

$$\big(T(3D+7)+3D+5\big)H^2+\big(T(4\boxed{d}+3D+10)+6\boxed{d}+3D+14\big)H+2$$

Spatial dimension

- Hyperparameters:

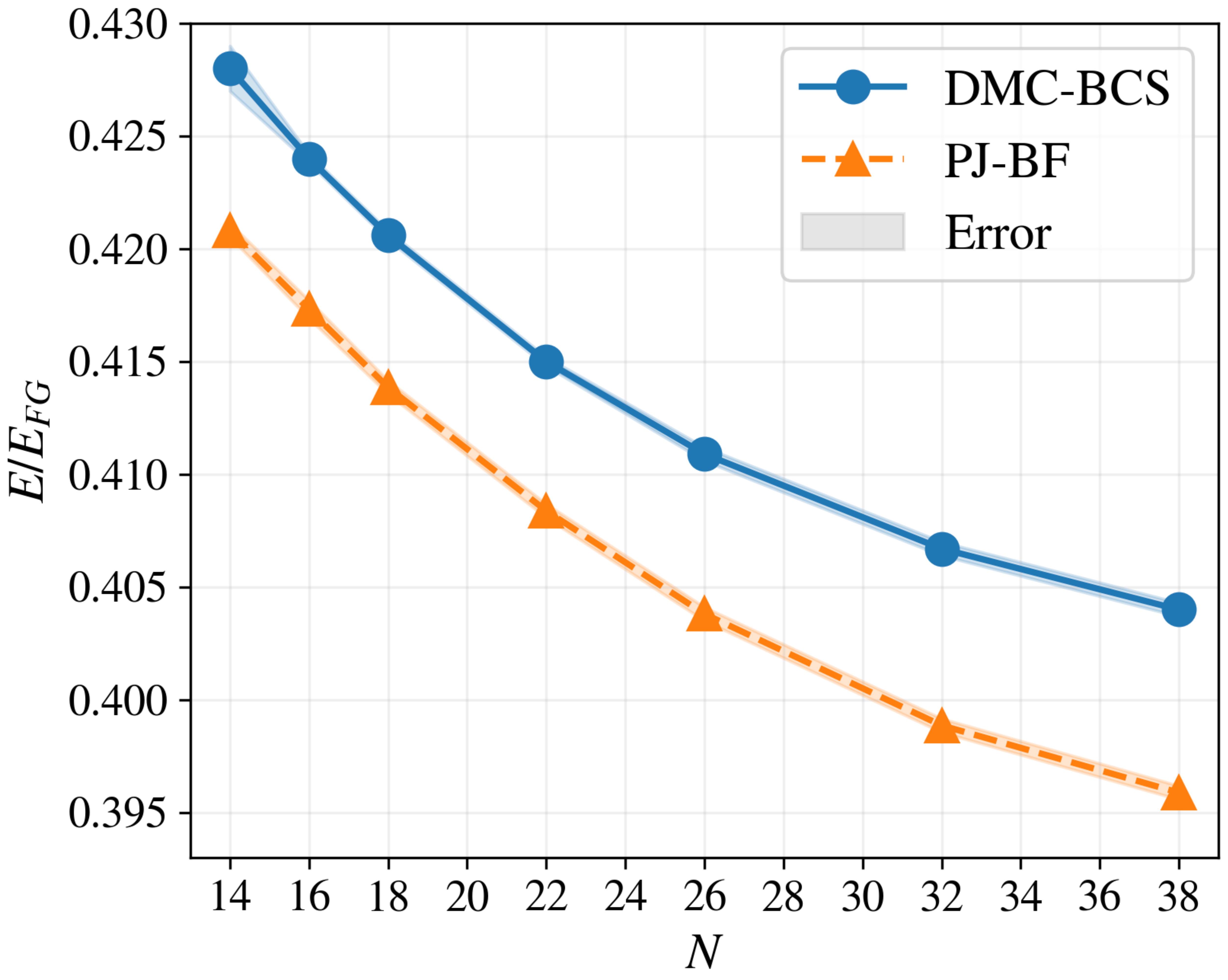
$T =$ Number of message-passing iterations

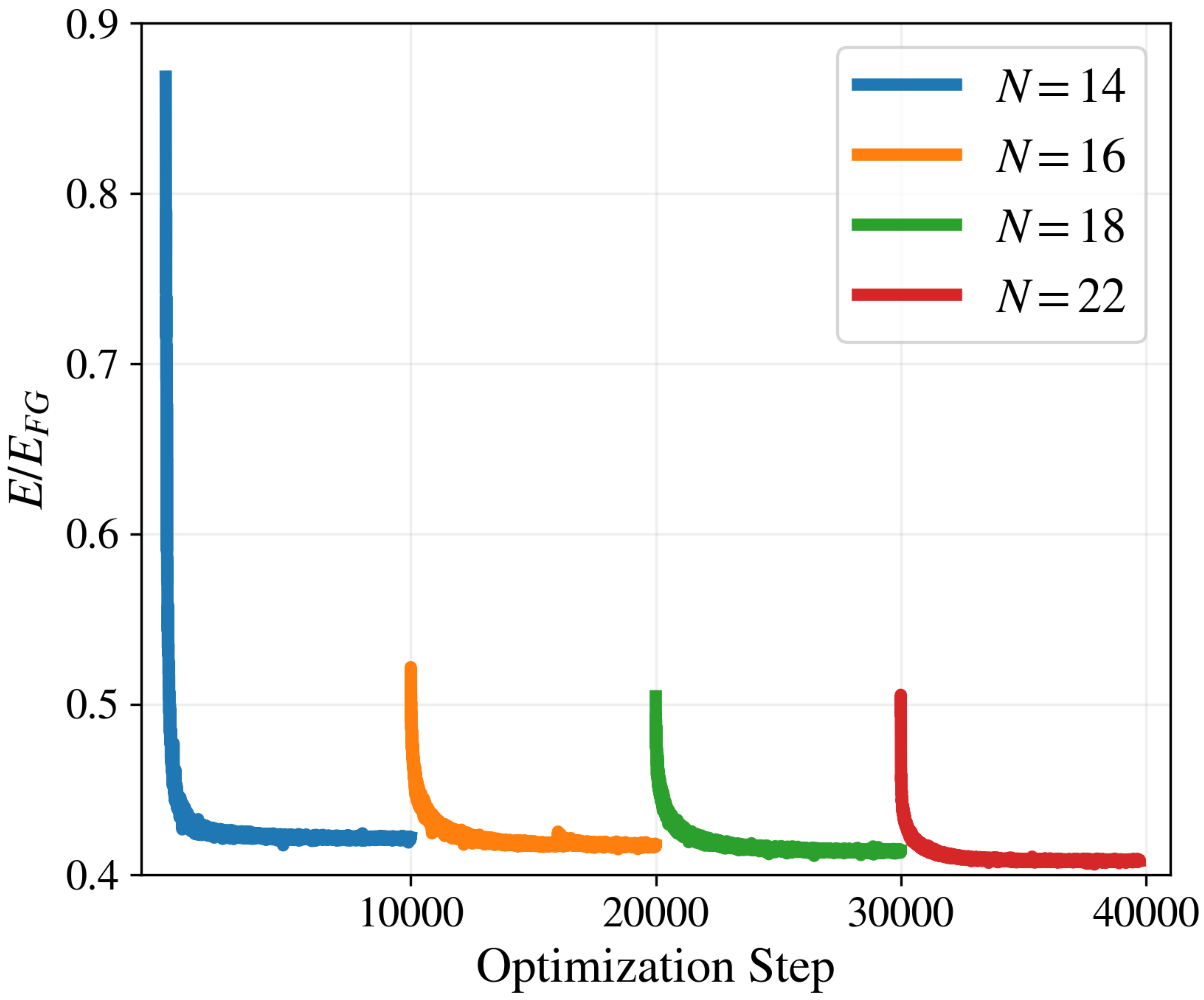
$D =$ Number of dense hidden layers in a single feedforward neural network

$H =$ Number of hidden nodes in a single dense layer

- Most of this work used $T=2,\ D=1,\ H=16 \quad \Longrightarrow \quad 8500$ parameters

MORE TRANSFER LEARNING

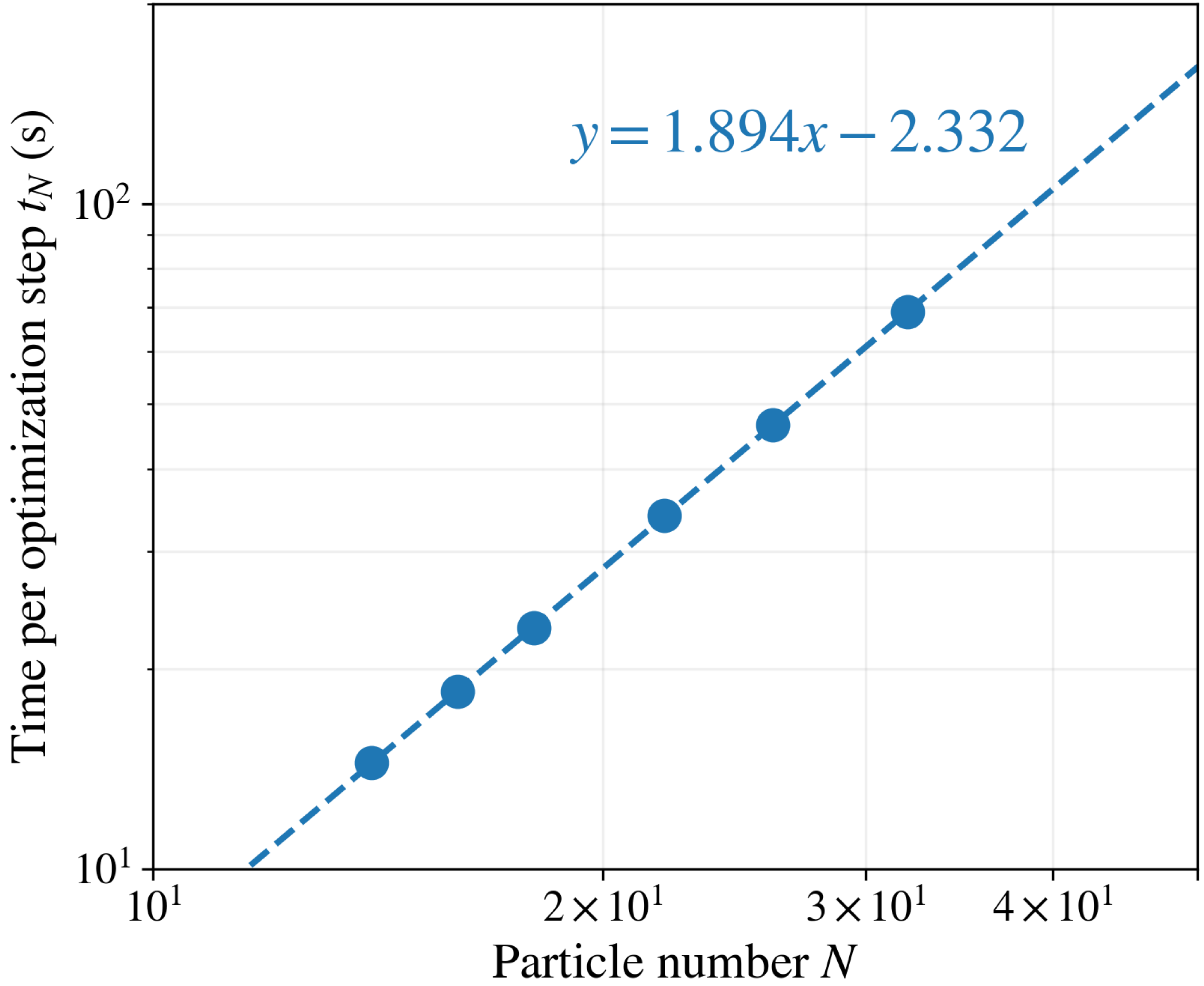




COMPUTATIONAL SCALING

Using 4 NVIDIA-A100s for $N \leq 32$

Empirical scaling $\mathcal{O}(N^{1.894})$



MODIFICATIONS FOR NUCLEAR SYSTEMS

- Change hamiltonian (we use a simple LO pionless EFT for now)
- Nuclear matter:
    - Add isospin τ_i into one-body input
    - Add products of isospins $\tau_i \cdot \tau_j$ into two-body input
- Nuclei:
    - Add positions without center-of-mass contribution $r_i - r_{cm}$ into one-body input
    - Add a confining Gaussian envelope
    - Remove periodic transformation of coordinates, distances, separation vectors

RECENT DEVELOPMENTS

communications physics

Article

https://doi.org/10.1038/s42005-024-01613-w

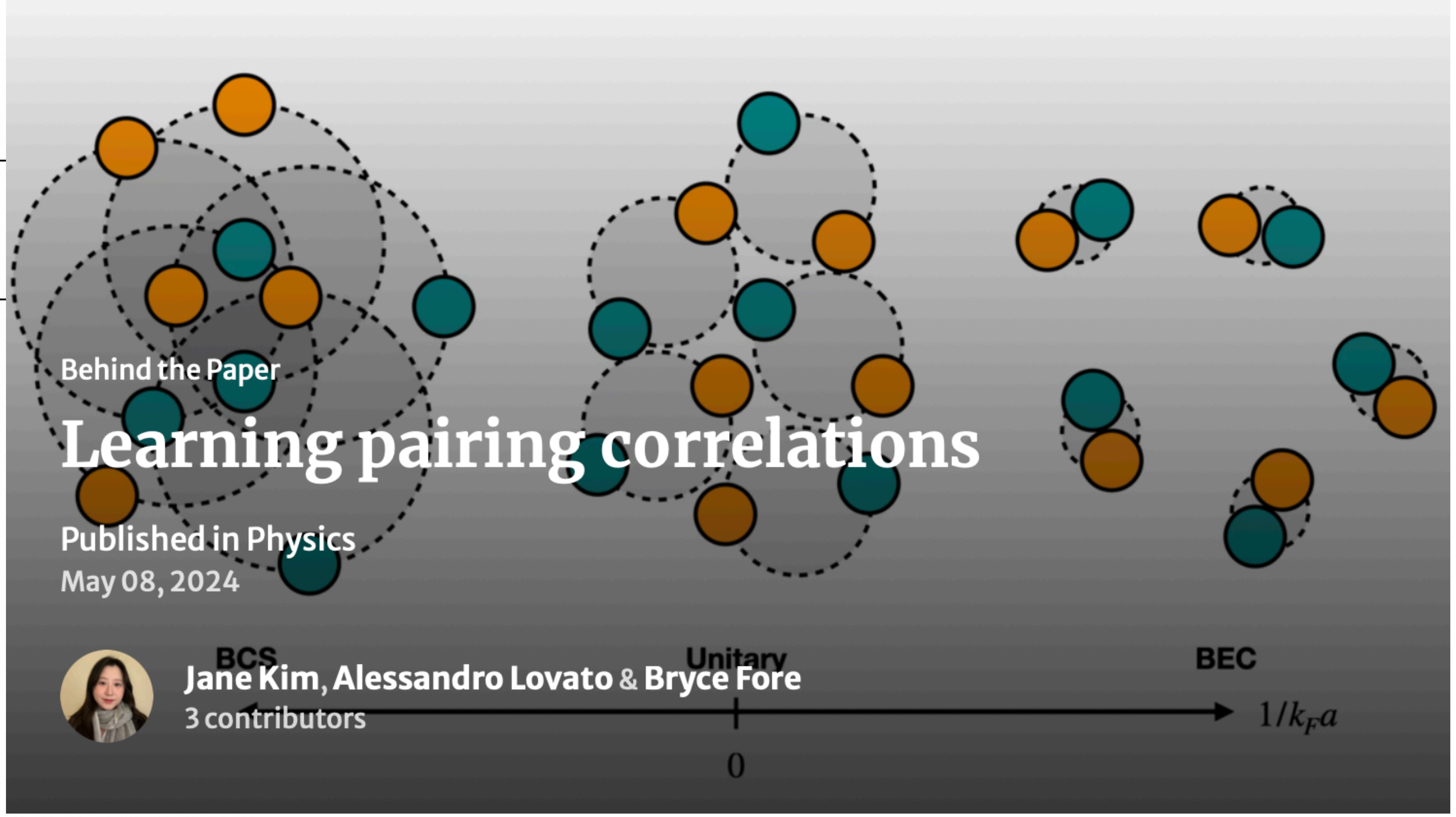
JK, MHJ, AL *et al. Commun. Phys.* 7, 148 (2024).

Neural-network quantum states for ultra-cold Fermi gases

Jane Kim[1,9], Gabriel Pescia[2,3], Bryce Fore[4], Jannes Nys[2,3], Giuseppe Carleo[2,3], Stefano Gandolfi[5], Morten Hjorth-Jensen[1,6] & Alessandro Lovato[4,7,8]

Ultra-cold Fermi gases exhibit a rich array of quantum mechanical properties, including the transition from a fermionic superfluid Bardeen-Cooper-Schrieffer (BCS) state to a bosonic superfluid Bose-Einstein condensate (BEC). While these properties can be precisely probed experimentally, accurately describing them poses significant theoretical challenges due to strong pairing correlations and the non-perturbative nature of particle interactions. In this work, we introduce a Pfaffian-Jastrow neural-network quantum state featuring a message-passing architecture to efficiently capture pairing and backflow correlations. We benchmark our approach on existing Slater-Jastrow frameworks and state-of-the-art diffusion Monte Carlo methods, demonstrating a performance advantage and the scalability of our scheme. We show that transfer learning stabilizes the training process in the presence of strong, short-ranged interactions, and allows for an effective exploration of the BCS-BEC crossover region. Our findings highlight the potential of neural-network quantum states as a promising strategy for investigating ultra-cold Fermi gases.

Behind the paper



ROUGH IDEAS?

EXCITED STATES

- Coupled stochastic reconfiguration?
  - Suppose we have two states $|\Psi_{\theta_1}\rangle,\ |\Psi_{\theta_2}\rangle$ that are orthogonal $\langle\Psi_{\theta_1}|\Psi_{\theta_2}\rangle = 0$
  - In previous VMC studies, they stay approximately orthogonal under imaginary-time propagation for a few iterations
  - Can we come up with an optimization scheme that preserved orthogonality during training?

$$\begin{bmatrix} S_{\theta_1} & \tilde{S}_{\theta_1,\theta_2} \\ \tilde{S}_{\theta_2,\theta_1} & S_{\theta_2} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 & \\ & \Delta\theta_2 \end{bmatrix} = -\frac{\Delta\tau}{2} \begin{bmatrix} \nabla_{\theta_1}\langle E_{\theta_1}\rangle & G_{\theta_1,\theta_2} \\ G_{\theta_2,\theta_1} & \nabla_{\theta_2}\langle E_{\theta_2}\rangle \end{bmatrix}$$

EXCITED STATES

- Coupled stochastic reconfiguration?
  - Suppose we have two states $|\Psi_{\theta_1}\rangle,\ |\Psi_{\theta_2}\rangle$ that are orthogonal $\langle\Psi_{\theta_1}|\Psi_{\theta_2}\rangle = 0$
  - In previous VMC studies, they stay approximately orthogonal under imaginary-time propagation for a few iterations
  - Can we come up with an optimization scheme that preserved orthogonality during training?

$$\begin{bmatrix} S_{\theta_1} & \tilde{S}_{\theta_1,\theta_2} \\ \tilde{S}_{\theta_2,\theta_1} & S_{\theta_2} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 & \\ & \Delta\theta_2 \end{bmatrix} = -\frac{\Delta\tau}{2} \begin{bmatrix} \nabla_{\theta_1}\langle E_{\theta_1}\rangle & G_{\theta_1,\theta_2} \\ G_{\theta_2,\theta_1} & \nabla_{\theta_2}\langle E_{\theta_2}\rangle \end{bmatrix}$$

Another factor of $\Delta\tau$

Cross expectations of $\hat{H}^2\ldots$

EXCITED STATES

- Diagonalizing the Hamiltonian in a subspace spanned by snapshots $\{\Psi_{\theta_1}, \Psi_{\theta_2}\}$
    - Obtain estimates of lowest-lying eigenstates and energies

$$\begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \Psi_{\theta_1} \\ \Psi_{\theta_2} \end{bmatrix}$$

- Match $|\Psi_{\theta_1+\Delta\theta_1}\rangle$ to Φ_i

$$\Longrightarrow \begin{bmatrix} S_{\theta_1} & \tilde{S}_{\theta_1,\theta_2} \\ \tilde{S}_{\theta_2,\theta_1} & S_{\theta_2} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 & \\ & \Delta\theta_2 \end{bmatrix} = \begin{bmatrix} G_{\theta_1} & \tilde{G}_{\theta_1,\theta_2} \\ \tilde{G}_{\theta_2,\theta_1} & G_{\theta_2} \end{bmatrix} \begin{bmatrix} c_{11}-1 & c_{21} \\ c_{12} & c_{22}-1 \end{bmatrix}$$

No $\hat{H}$ appear…

EXCITED STATES

- Diagonalizing the Hamiltonian in a subspace spanned by snapshots $\{\Psi_{\theta_1}, \Psi_{\theta_2}\}$
  - Obtain estimates of lowest-lying eigenstates and energies

$$\begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \Psi_{\theta_1} \\ \Psi_{\theta_2} \end{bmatrix}$$

  - Match $|\Psi_{\theta_1+\Delta\theta_1}\rangle$ to Φ_i

Could also just maximize overlap the usual way

$$\Longrightarrow \begin{bmatrix} S_{\theta_1} & \tilde{S}_{\theta_1,\theta_2} \\ \tilde{S}_{\theta_2,\theta_1} & S_{\theta_2} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 & \\ & \Delta\theta_2 \end{bmatrix} = \begin{bmatrix} G_{\theta_1} & \tilde{G}_{\theta_1,\theta_2} \\ \tilde{G}_{\theta_2,\theta_1} & G_{\theta_2} \end{bmatrix} \begin{bmatrix} c_{11}-1 & c_{21} \\ c_{12} & c_{22}-1 \end{bmatrix}$$

No $\hat{H}$ appear…

EXCITED STATES

- How to pick initially orthogonal states $|\Psi_{\theta_1}\rangle,\ |\Psi_{\theta_2}\rangle$?
  - Diagonalize Hamiltonian in subspace spanned by some snapshots $\Longrightarrow$ maximize $\langle \Phi_i | \Psi_{\theta_i} \rangle$
  - If that fails… minimize overlap $\langle \Psi_{\theta_1} | \Psi_{\theta_2} \rangle$
  - If that fails… Gram-Schmidt

NQS IN MOMENTUM SPACE

- We should try this because….
    - Kinetic energy becomes trivial
    - For bound states, very minimal changes to NQS (just change inputs)
    - (I think) no one has done this yet $\implies$ some benefits we can't see yet?
    - Much more natural way of handling scattering problems variationally??
    - The pfaffian in momentum space

NQS IN MOMENTUM SPACE

- We should try this because….
  - Kinetic energy becomes trivial
  - For bound states, very minimal changes to NQS (just change inputs)
  - (I think) no one has done this yet $\implies$ some benefits we can't see yet?
  - Much more natural way of handling scattering problems variationally??
  - The pfaffian in momentum space

Short-range correlation physics

Anthony Tropiano, Dick Furnstahl, Scott Bogner

SCATTERING (KOHN)

- In position space, the scattering wave function is finite everywhere, but extends to ∞
- In momentum space, the scattering wave function has poles, but does not extend to ∞
- Radial equation ($\ell = 0$) in momentum space:

$$(p^2 - k^2)\varphi(p) + \int_0^\infty V(p, p')\varphi(p')dp' = 0$$

- Formal solution

Parameterize this with a neural network

$$\varphi(p) = \delta(p - k) - \frac{1}{p^2 - k^2}\int_0^\infty V(p, p')\varphi(p')dp' = \delta(p - k) - \frac{F(p)}{p^2 - k^2}$$

SCATTERING (KOHN)

- The pole would surely cause numerical instabilities $\Longrightarrow$ Regularize

e.g. $$\frac{1}{p^2 - k^2} \mapsto \frac{\tanh^2\left(\frac{p-k}{\epsilon}\right)}{p^2 - k^2}$$ for small $\epsilon > 0$

- Enforce boundary conditions with fixed Gaussian envelope $\exp\left(-\frac{p^2}{2\Lambda^2}\right)$

SCATTERING (KOHN)

- Find stationary solution of functional

$$J[\varphi] \equiv \int_0^\infty \varphi(p)\left[(p^2 - k^2)\varphi(p) + \int_0^\infty V(p,p')\varphi(p')dp'\right]dp$$

- Where

$$\phi_\epsilon(p) = \delta(p-k) - \frac{\tanh^2\left(\frac{p-k}{\epsilon}\right)\exp\left(-\frac{p^2}{2\Lambda^2}\right)}{p^2-k^2}$$

- Then use transfer learning

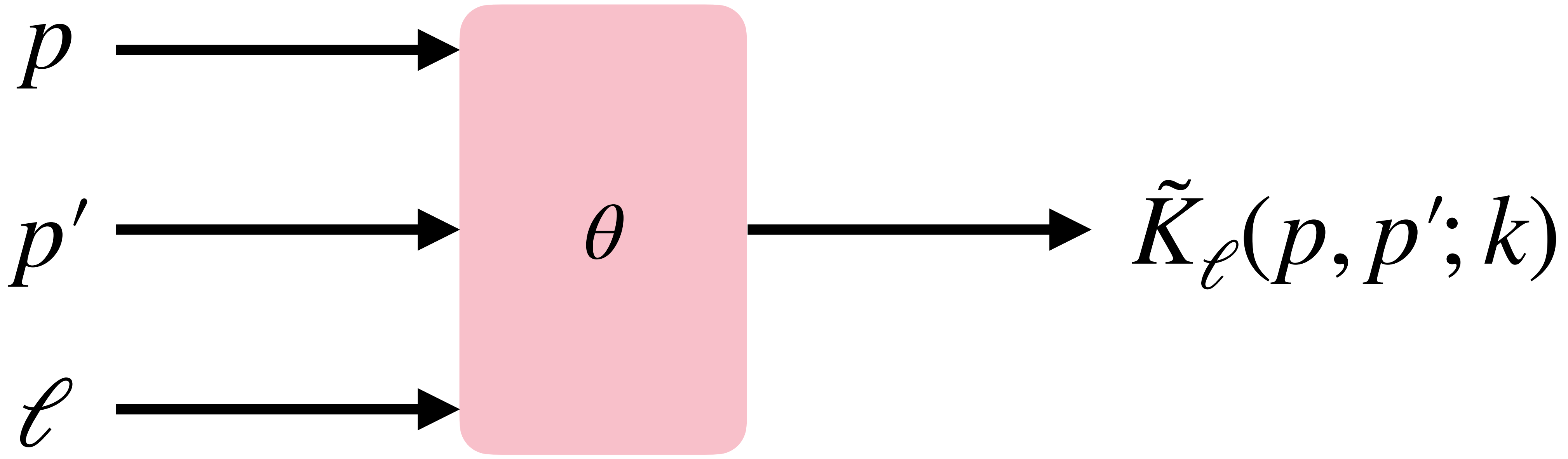
$$\lim_{\epsilon\to 0} J[\varphi_\epsilon] = J[\varphi] + 2F(k) = -\frac{2k}{\pi}\tan\delta(k)$$

SCATTERING (NEWTON)

- Lippmann-Schwinger equation

$$K = V + VG_0K \implies K = (I - VG_0)^{-1}V$$

- Ansatz $\tilde{K} \implies \mathscr{K}[\tilde{K}] = V + VG_0\tilde{K} + \tilde{K}G_0V - \tilde{K}G_0\tilde{K} + \tilde{K}G_0VG_0\tilde{K}$



- Since we parameterize $\tilde{K}(\theta)$ by a neural network, we can be more flexible with the inputs (e.g. add parameters of the potential too?)

THANK YOU!