

Support informations for Solving the Hubbard model with Neural Quantum States

Yuntian Gu et al.

Supplementary Note 1. Pretrain

Instead of randomly initializing the NQS ansatz, we use a pretrain method to provide the transformer wavefunction better initial parameters before optimizing it with MARCH.

We first train a simple neural network similar to neural network backflow (NNB) [1]. A multilayer perceptron (MLP) with two hidden layers maps the occupation number $\mathbf{n} \in \{0, 1\}^{2N}$ to a orbital matrix $M_{\text{nnb}}(\mathbf{n}) \in \mathbb{R}^{2N \times N_e}$, which is restricted to a block-diagonal matrix,

$$M_{\text{nnb}}(\mathbf{n}) = \begin{pmatrix} M_{\text{nnb}}^{\uparrow}(\mathbf{n}) & 0 \\ 0 & M_{\text{nnb}}^{\downarrow}(\mathbf{n}) \end{pmatrix}, \quad (1)$$

where the $M_{\text{nnb}}^{\uparrow}(\mathbf{n}), M_{\text{nnb}}^{\downarrow}(\mathbf{n}) \in \mathbb{R}^{N \times N_e/2}$ are the orbitals of spin-up and spin-down electrons, respectively. Based on the occupation number, N_e rows of orbital matrix $M_{\text{nnb}}(\mathbf{n})$ are extracted to form a square orbital matrix $\Phi \in \mathbb{R}^{N_e \times N_e}$, which is also block-diagonal,

$$\Phi = \begin{pmatrix} \Phi^{\uparrow} & 0 \\ 0 & \Phi^{\downarrow} \end{pmatrix}. \quad (2)$$

The amplitude of NNB ansatz is defined by the determinant of a matrix Φ :

$$\psi_{\text{nnb}}(\mathbf{n}) = \det[\Phi]. \quad (3)$$

This ansatz is variationally optimized using the VMC method, with the hyperparameters specified in Table 5. Subsequently, the optimized NNB ansatz is used to generate a labeled dataset, S_{pre} , of size N_{pre} :

$$S_{\text{pre}} = \{\mathbf{n}_i, M_{\text{nnb}}(\mathbf{n}_i)\}_{i=1}^{N_{\text{pre}}}, \quad (4)$$

where each entry consists of a configuration $\mathbf{n}_i$ and its corresponding orbital matrix $M_{\text{nnb}}(\mathbf{n}_i)$.

The transformer ansatz is then pretrained in a supervised manner, with the loss function to be the mean square error of the orbital matrix, i.e.,

$$\mathcal{L}_{\text{pretrain}} = \frac{1}{N_{\text{pre}}} \sum_{i=1}^{N_{\text{pre}}} \sum_{k=1}^K ||M^k(\mathbf{n}_i) - M_{\text{nnb}}(\mathbf{n}_i)||^2, \quad (5)$$

with Adam optimizer. We note that in practice, the pretrain dataset can be generated online. Compared to random initialization, a pretrained transformer is much more stable and converges significantly faster.

Supplementary Note 2. Pinning Field

To avoid convergence to local minima during optimization, a temporary pinning field is applied. This field takes the form of an antiferromagnetic (AFM) term added to the Hamiltonian $\hat{H}$ on specific columns:

$$\hat{H}_{\text{pin}} = \hat{H} + h_m \sum_{i \in [L_x]} h_i \sum_{j \in [L_y]} (-1)^j \hat{S}_{ij}^z, \quad (6)$$

with $h_m = 0.2$ and $h_i \in \{-1, 0, 1\}$ to probe the possible magnetic order. Initially, a transformer wavefunction ψ_1 is optimized under the influence of this pinning field. This pinned field guides the ansatz to fall into a desired pattern. Subsequently, a second transformer ansatz ψ_2 is initialized. We then employ a pretraining procedure aimed at maximizing the fidelity $\frac{\langle \psi_1 | \psi_2 \rangle^2}{\langle \psi_1 | \psi_1 \rangle \langle \psi_2 | \psi_2 \rangle}$. This effectively transfers the desired pattern to ψ_2 . The final optimization is then performed on ψ_2 with the pinning field removed, allowing it to relax and find the true ground state.

With this method, we can stabilize a vertical or horizontal stripe. From Figure 31 we can see that at the same network size, the vertical stripe is higher in energy than the horizontal stripe, indicating the horizontal stripe indeed is favored on 32×8 lattice with $t' = -0.2$ and $\delta = 1/8$.

Supplementary Table 1 | Benchmark energy in pure Hubbard model at half-filling with PBC. NQS achieve comparable results with the numerically exact AFQMC values [2].

Systems	8×8	10×10	12×12
AFQMC	-0.5262(5)	-0.5254(3)	-0.5246(1)
NQS	-0.52582	-0.52492	-0.52440

Supplementary Table 2 | Benchmark energy in pure Hubbard model with OBC. NQS achieve new state-of-the-art results by surpassing PEPS.

Systems	16×4	16×6	8×8	16×8	16×12	16×16
DMRG	-0.68537	-0.70550	-0.69840	-0.71250		
PEPS	-0.68304(5)	-0.7008(2)	-0.69928(3)	-0.7122(3)	-0.7202(2)	-0.7260(2)
NQS	-0.68325	-0.70307	-0.69952	-0.71309	-0.72212	-0.72747
Hartree-Fock	-0.52499	-0.54269	-0.53874	-0.55335	-0.56250	-0.56716

Supplementary Note 3. Benchmark Results

In addition to the results presented for the pure Hubbard model with OBC in the main text (numbers are listed in Table 2), this section details extensive benchmark comparisons of our NQS methodology against several leading numerical techniques across a variety of Hubbard model configurations and system parameters. We also provide an ablation study for our proposed optimizer MARCH against SPRING.

Supplementary Note 3.1 Comparison with Hidden Fermion Determinantal State

We first benchmark our NQS approach against the Hidden Fermion Determinantal States (HFDS) method, a notable prior NQS implementation [5]. For the pure Hubbard model on a 16×4 lattice with PBC, which represents the largest calculation reported in their paper, our NQS yields a ground state energy of -0.76298 . This value is significantly lower than the HFDS variational energy of $-0.753(2)$.

Supplementary Note 3.2 Comparison with DMRG

We further compare our NQS method with the DMRG on the t' Hubbard model. Specifically, for a cylindrical geometry 16×8 with next-nearest-neighbor hopping $t' = -0.2$, our NQS achieves a variational energy of -0.72791 . This is lower than the DMRG energy of -0.72747 , obtained with a large bond dimension of $m = 40000$.

Supplementary Note 3.3 Comparison with PEPS

We also benchmark the computational efficiency of NQS against PEPS on identical hardware. The implementation of PEPS can be referred to Wu et al. [7]. As detailed in Figure 32, the expressive power of NQS is similar to fermion PEPS with $D = 10$. However, NQS is significantly better than PEPS in terms of computational efficiency under the same hardware conditions, as shown in Table 3.

Supplementary Note 3.4 Optimizer Ablation Study

To assess the efficacy of our proposed optimizer MARCH, we performed an ablation study comparing its performance against SPRING [8] and Min-SR [9, 10]. The convergence behavior of both optimizers during the NQS training is presented in Figure 33. These results clearly demonstrate that MARCH achieves significantly faster and more stable convergence compared to SPRING and Min-SR for the systems tested.

Supplementary Note 3.5 Accuracy Estimation

To compare the accuracy of different NQS, we benchmark them on relatively simple systems with known exact solutions. Specifically, we used a 4×4 lattice or the half-filled pure Hubbard model, which can be solved exactly using Exact Diagonalization or AFQMC. The results are presented in Table 4.

Supplementary Table 3 | Computation complexity relative to NQS under the same hardware condition in 8×8 lattice. NQS is significantly more efficient than PEPS.

Methods	NQS	NQS large	PEPS D=8	PEPS D=10
Wall Time	1×	2×	7×	24×

Supplementary Table 4 | Accuracy on simple systems, where the exact solution is possible.

Algorithm	[3]	[1]	[4]	[5]	[6]	This work
Lattice	8×8	4×4	4×4	4×4	8×8	12×12
Dopping	0	1/8	3/8	1/8	0	0
Relative Error	0.003	0.01	0.0005	0.001	0.005	0.0002
Accuracy	99.7%	99%	99.95%	99.9%	99.5%	99.98%

Supplementary Note 4. Hyperparameter

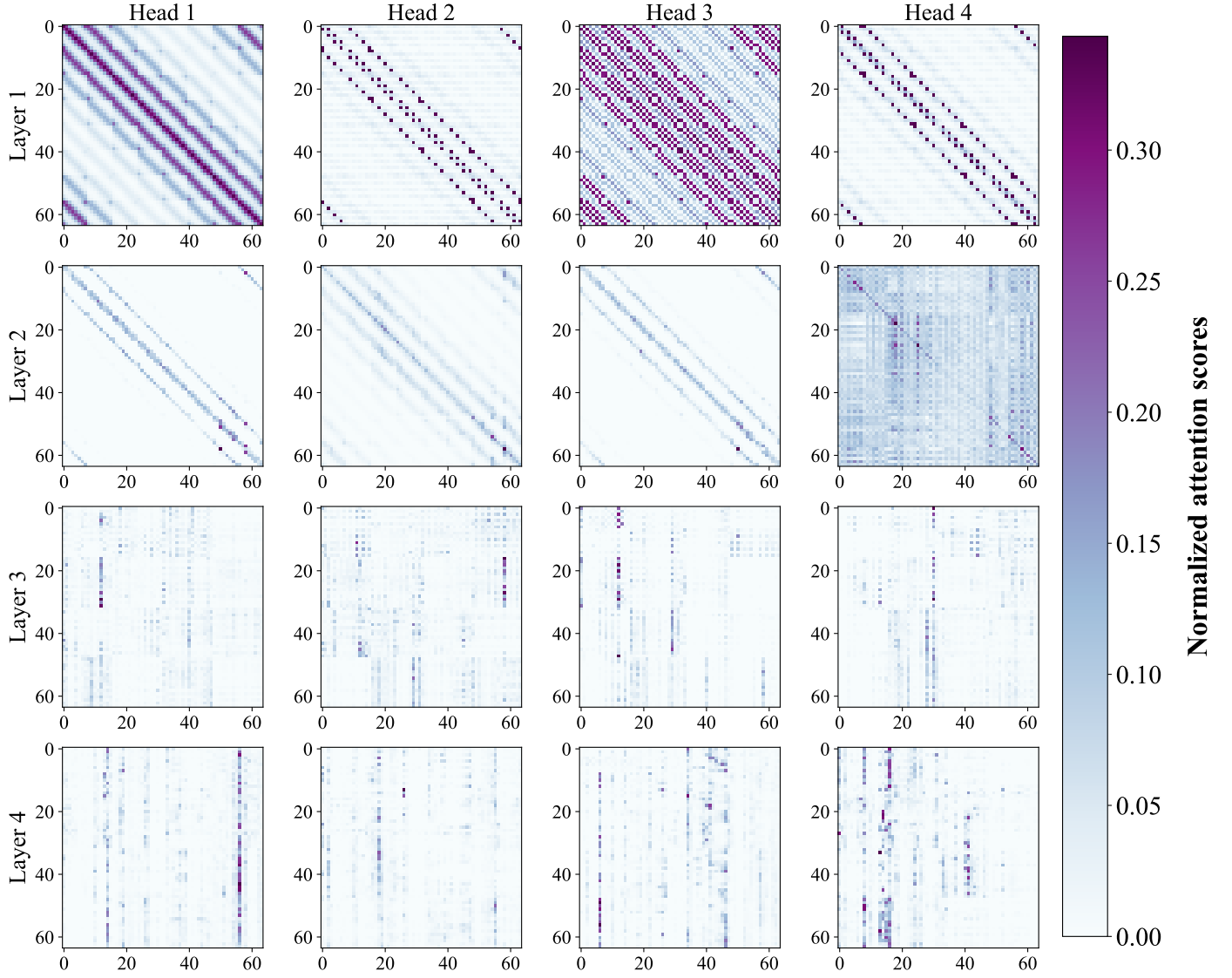
Config	Value
Optimizer	Adam
Optimizer momentum	$\mu, \beta = 0.9, 0.999$
Batch size	4096
Learning rate at time t	$10^{-4}(1 + t/10^4)^{-1}$
Local energy clipping	5.0
MCMC step	$2.5L_x \times L_y$
Hidden dimension	256
Layers	2
Steps	20000

Supplementary Table 5 | List of configs for the NNB training.

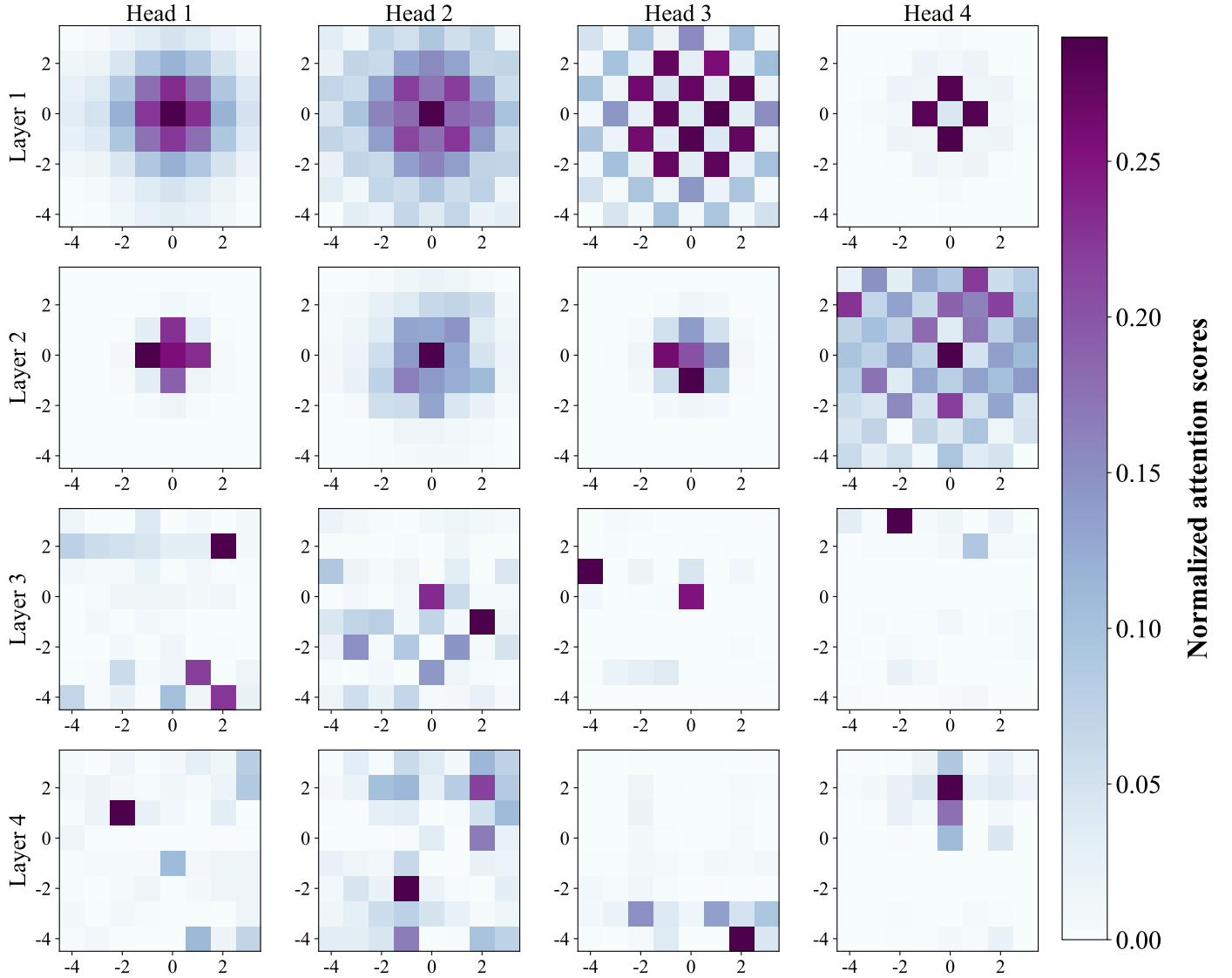
Config	Value
Optimizer	Adam
Batch size	4096
Learning rate	3×10^{-4}
MCMC step	30
Steps	5000

Supplementary Table 6 | List of configs for the pretraining phase.

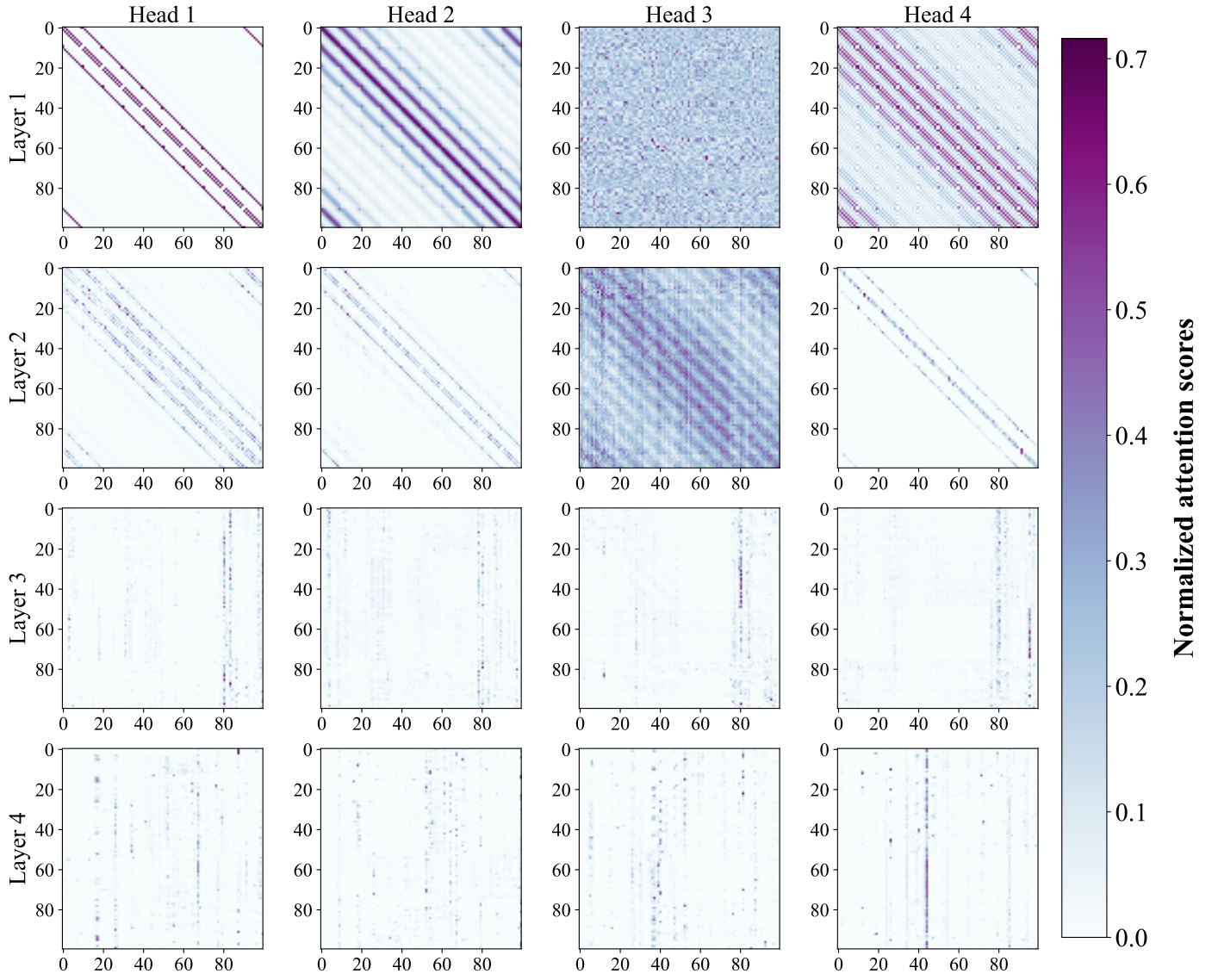
We present general settings of NNB training, pretraining, and MARCH training in Table 5, Table 6, and Table 7, respectively. We note that for different systems, the hyperparameters might be slightly different. All code and data will be made openly available upon publication.



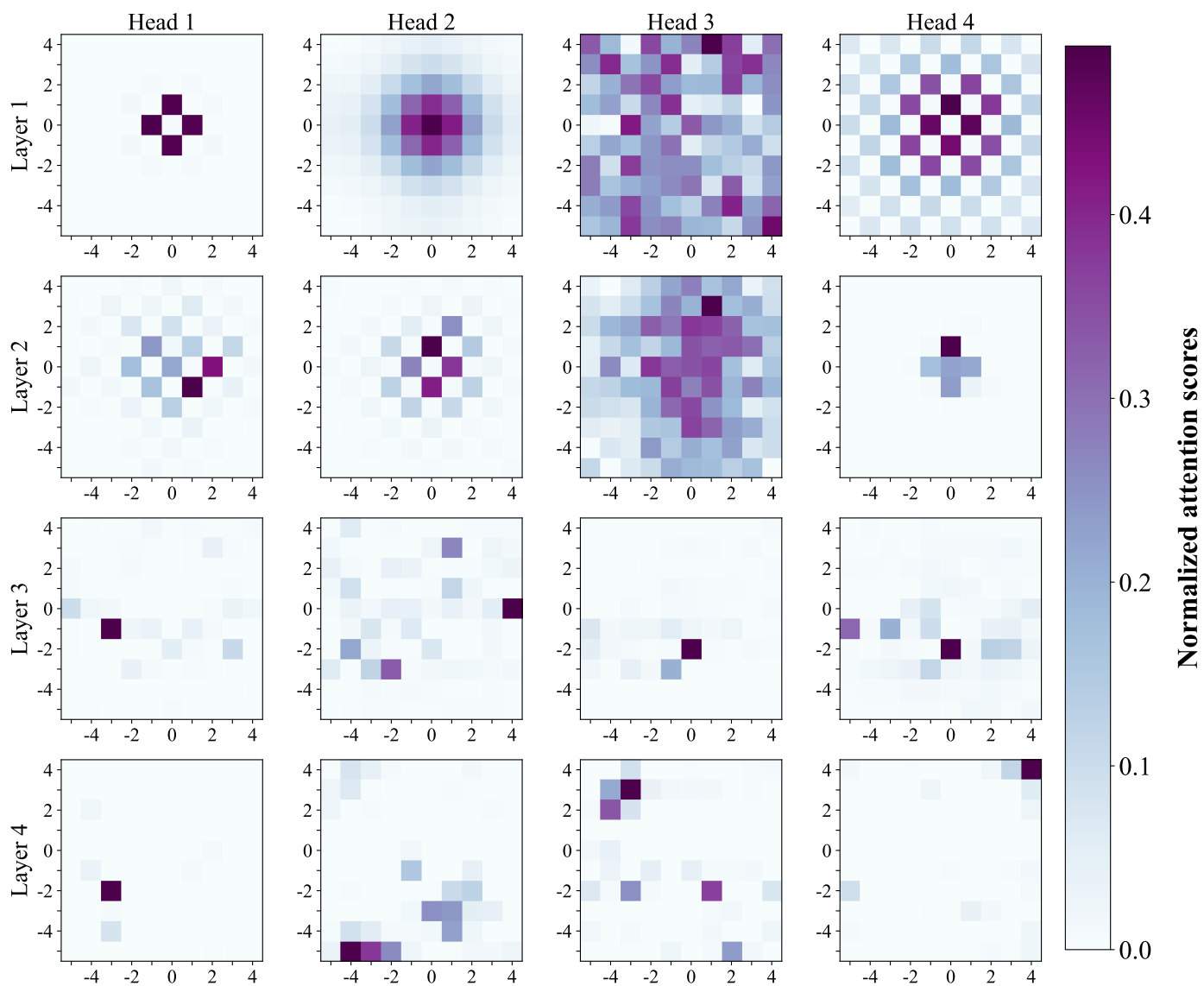
Supplementary Figure 1 | Attention maps for the transformer average over the sample electron configuration for 8×8 pure Hubbard model at half-filling. The maps are organized by network layer (top to bottom: Layer 1 to Layer 4) and attention head (left to right: Head 1 to Head 4).



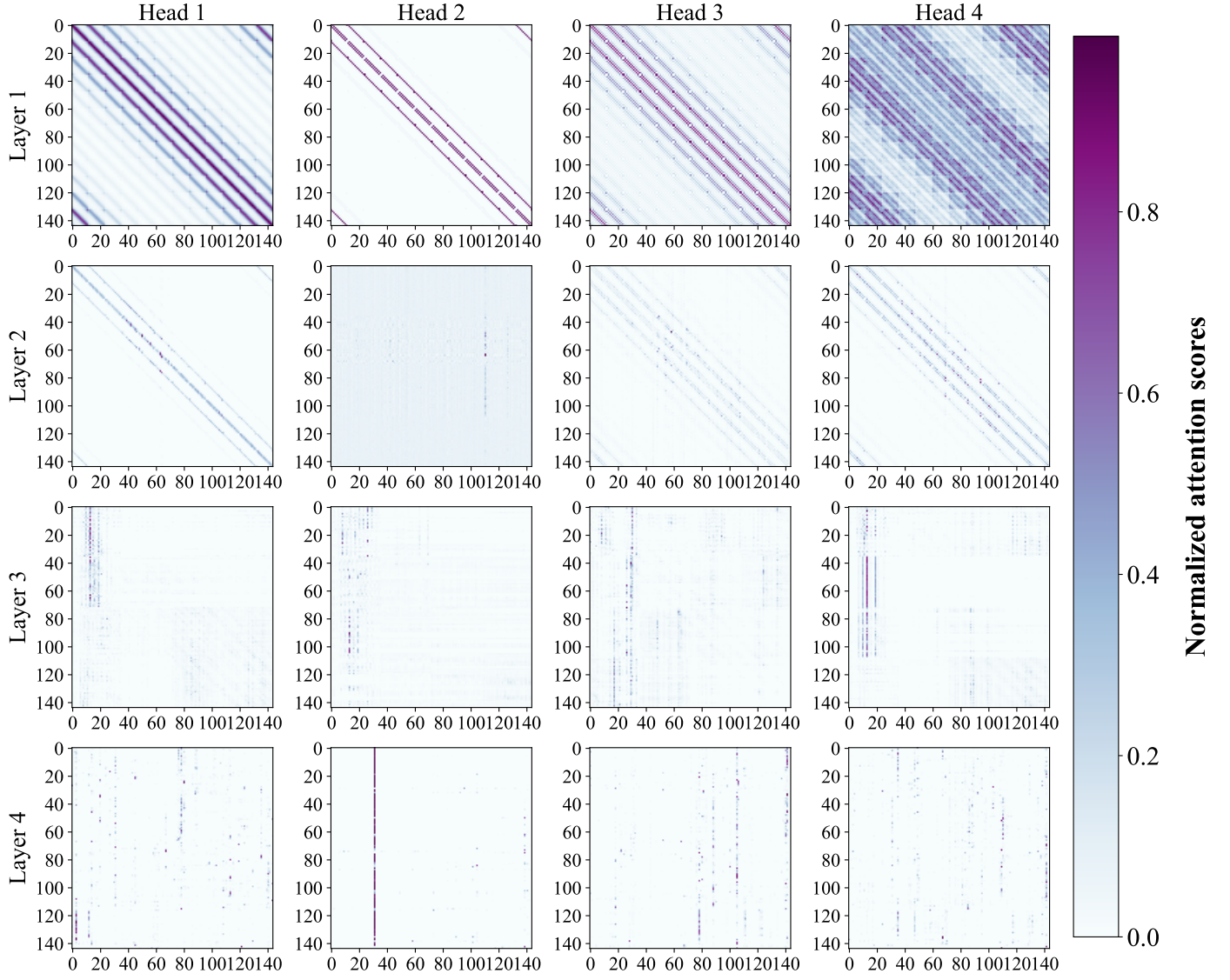
Supplementary Figure 2 | Visualization of the attention mechanism, showing how a lattice site attends to another target site for 8×8 pure Hubbard model at half filling. The maps are organized by network layer (top to bottom: Layer 1 to Layer 4) and attention head (left to right: Head 1 to Head 4).



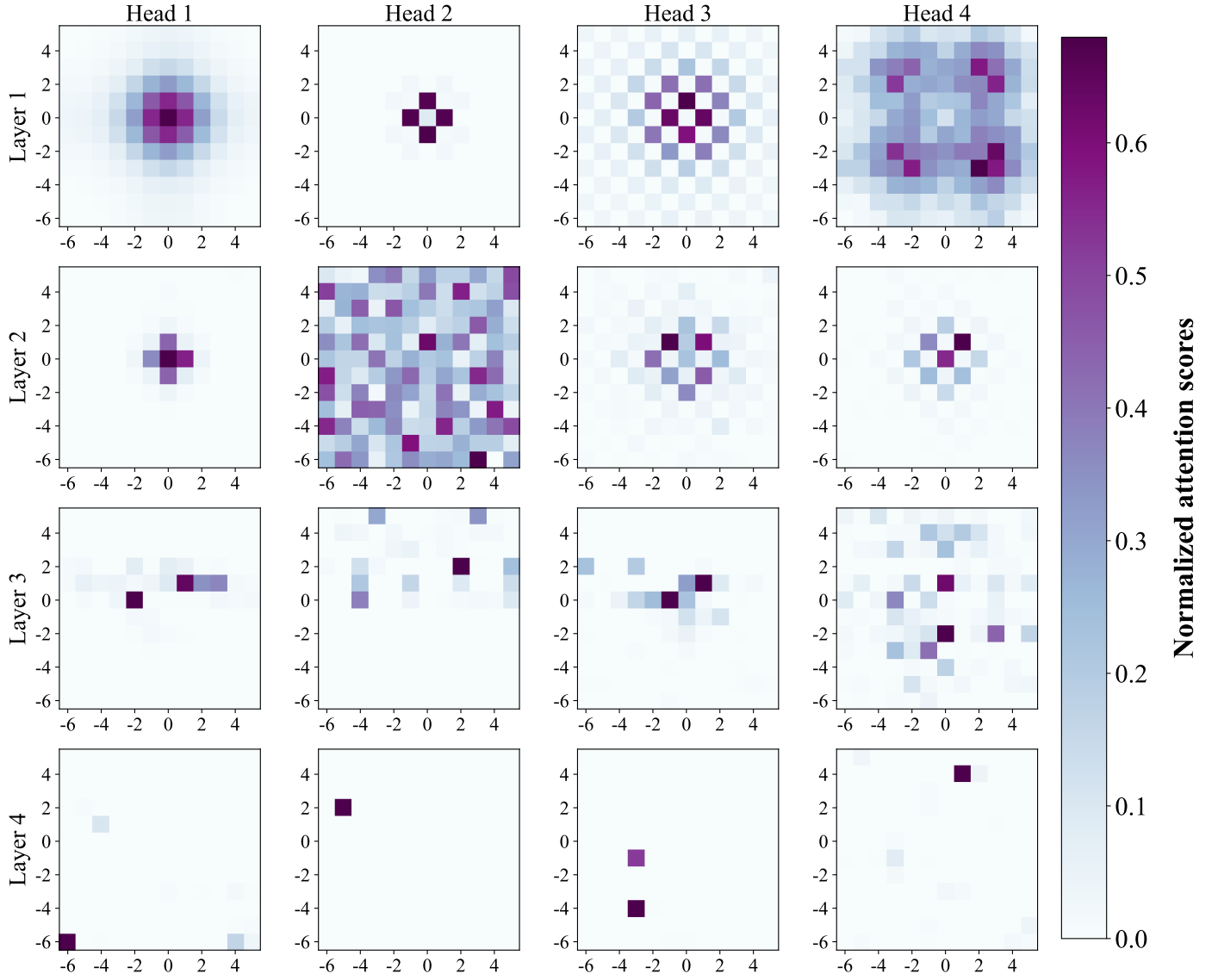
Supplementary Figure 3 | Similar as Figure 1 but for 10×10 pure Hubbard model at half-filling.



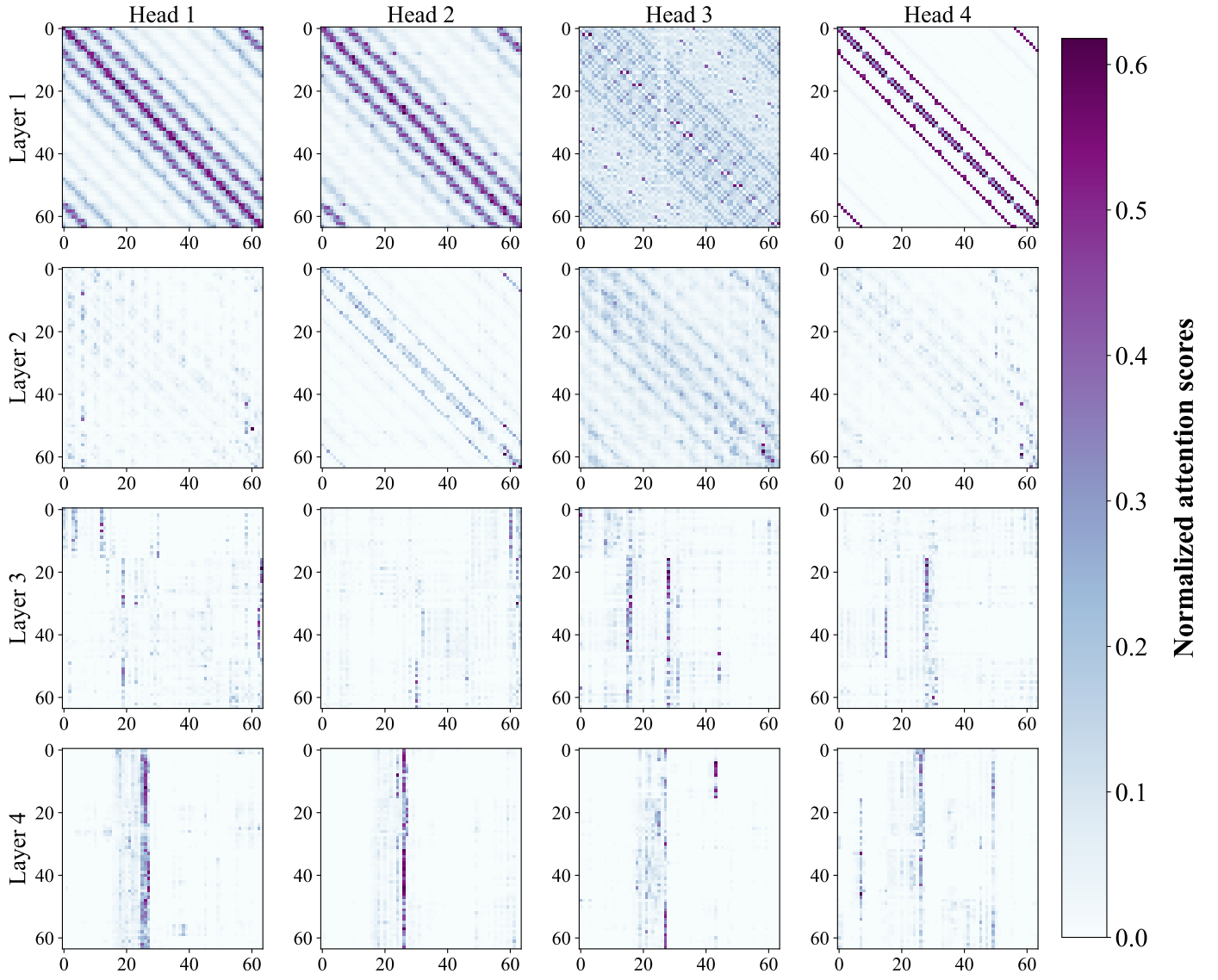
Supplementary Figure 4 | Similar as Figure 2 but for 10×10 pure Hubbard model at half-filling.



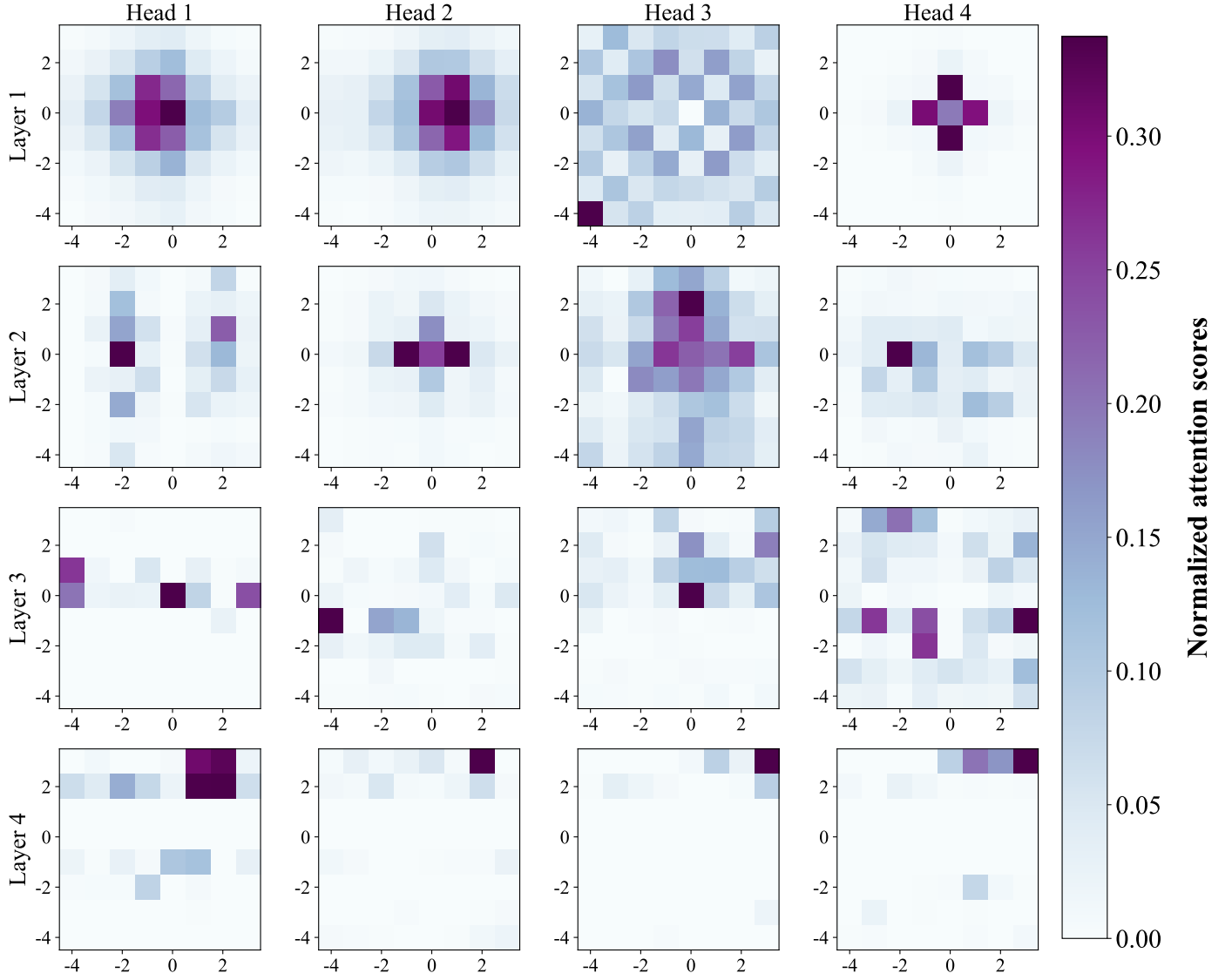
Supplementary Figure 5 | Similar as Figure 1 but for 12×12 pure Hubbard model at half-filling.



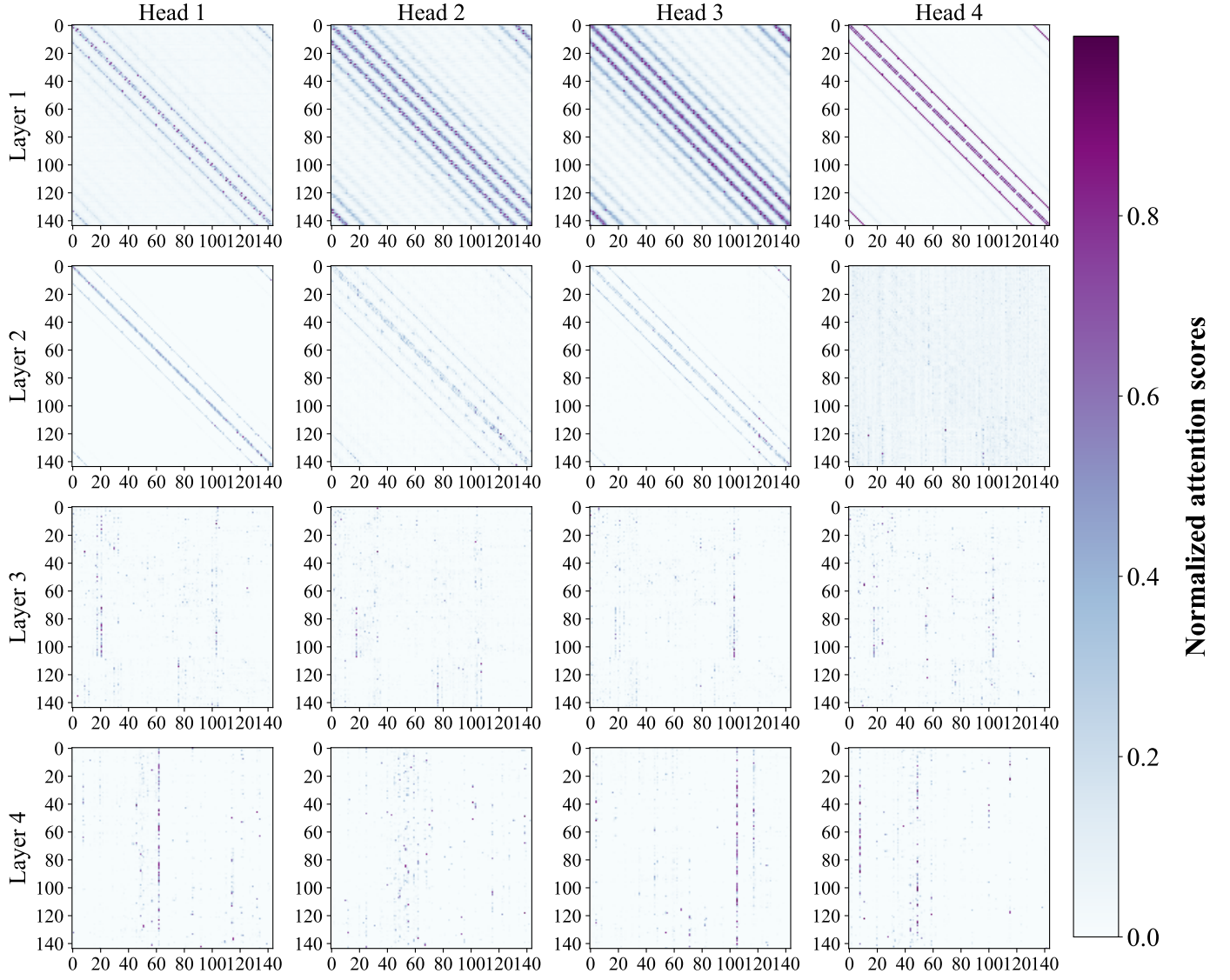
Supplementary Figure 6 | Similar as Figure 2 but for 12×12 pure Hubbard model at half-filling.



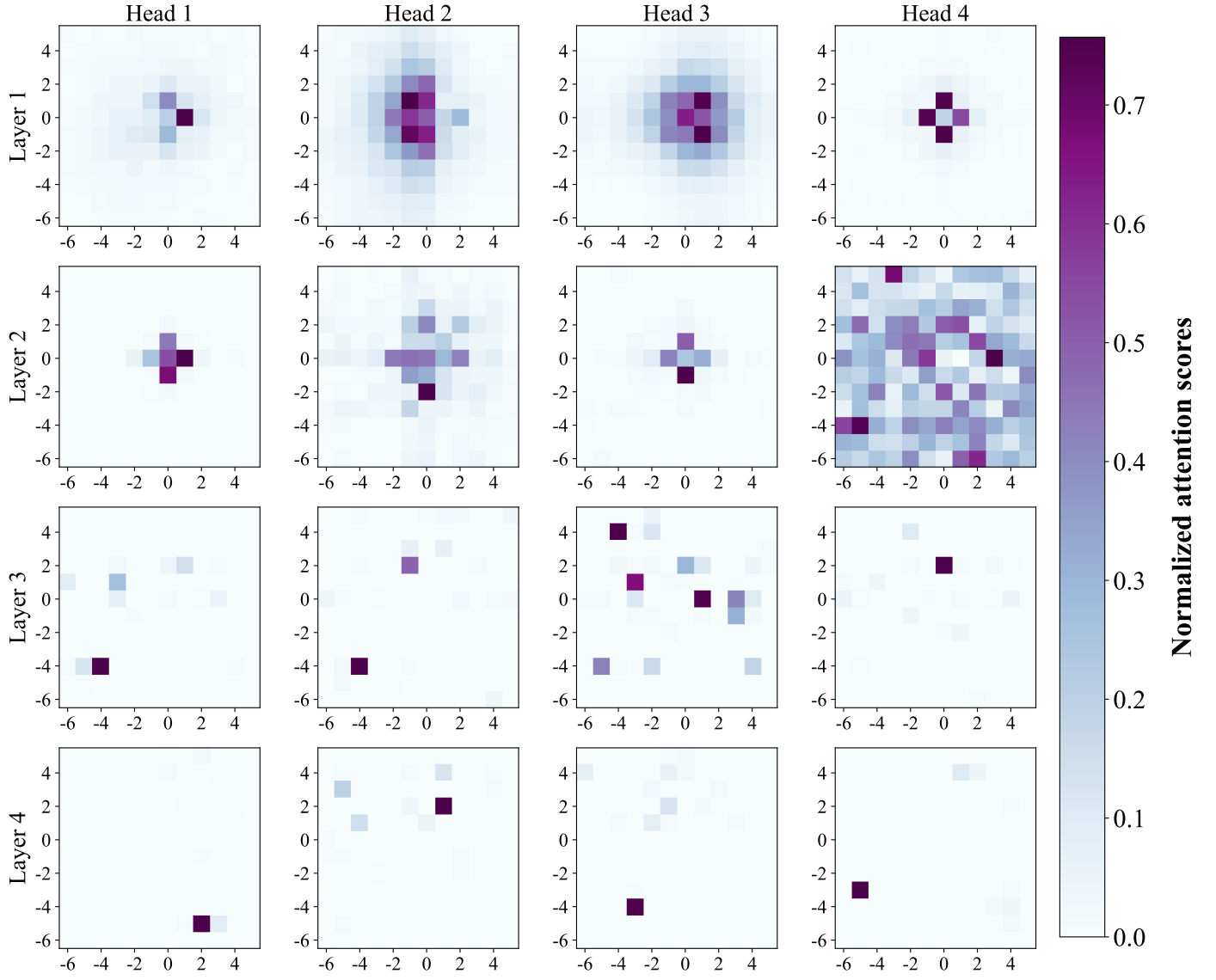
Supplementary Figure 7 | Similar as Figure 1 but for 8×8 Hubbard model with $t' = -0.2$ at 1/8 doping.



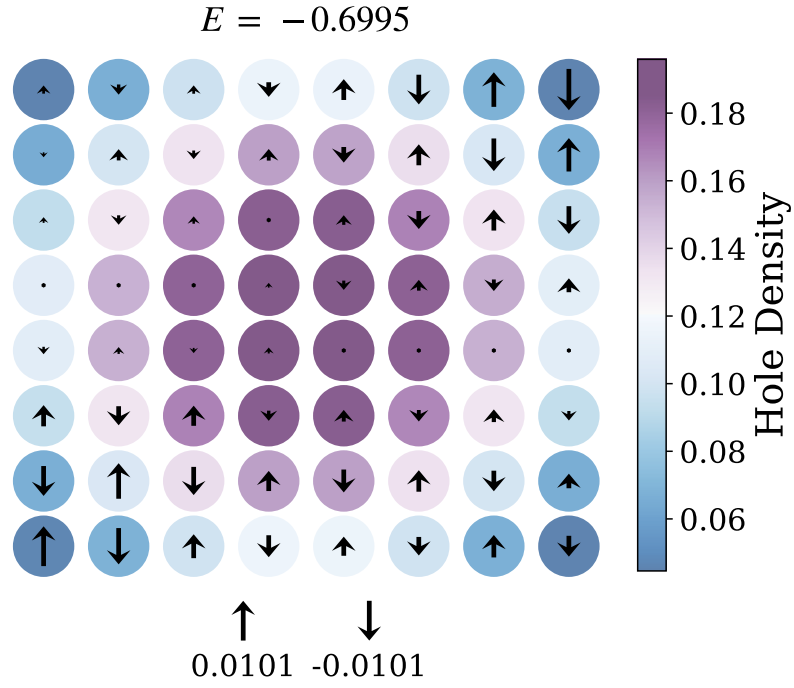
Supplementary Figure 8 | Similar as Figure 2 but for 8×8 Hubbard model with $r' = -0.2$ at 1/8 doping.



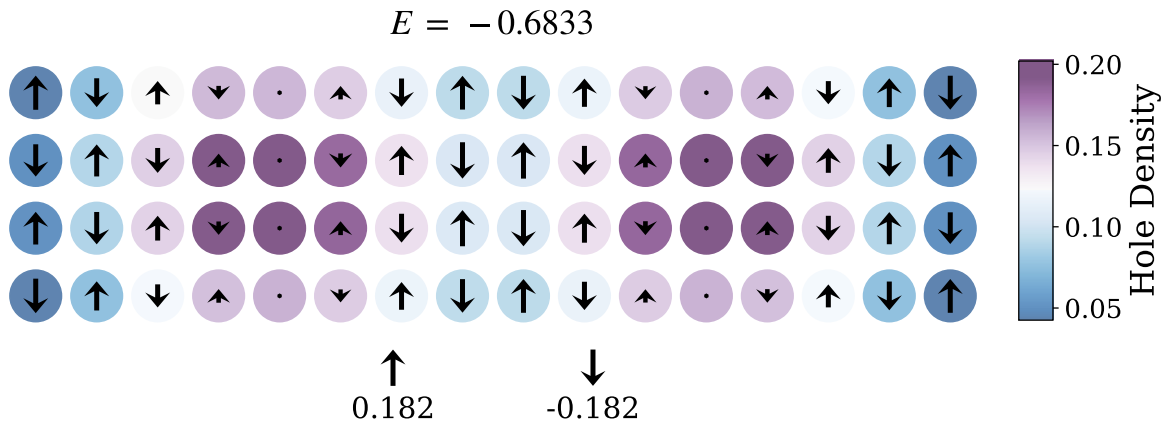
Supplementary Figure 9 | Similar as Figure 1 but for 12×12 Hubbard model with $t' = -0.2$ at $1/8$ doping.



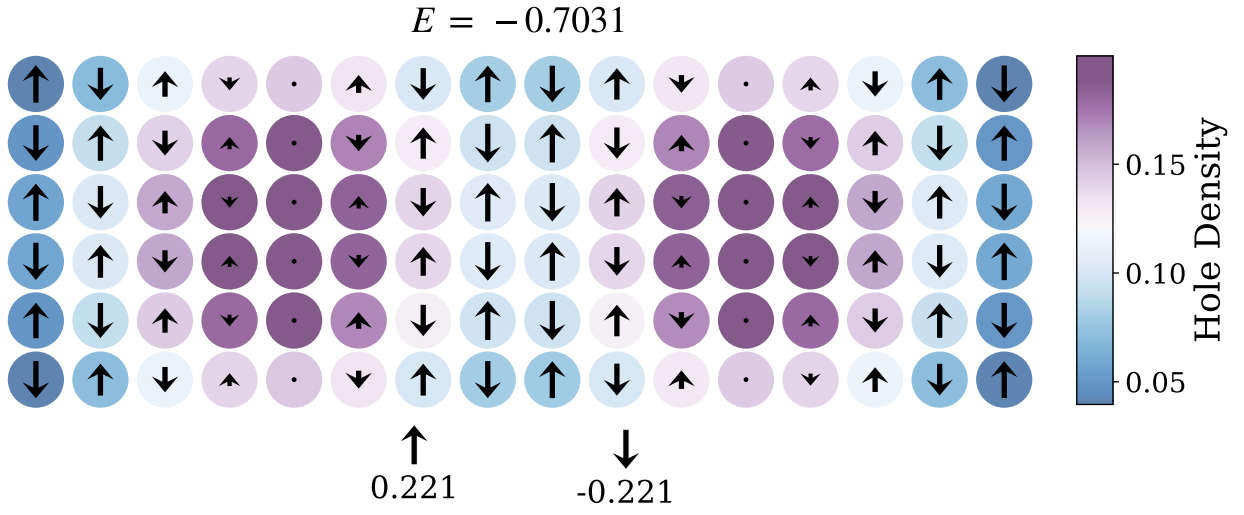
Supplementary Figure 10 | Similar as Figure 2 but for 12×12 Hubbard model with $t' = -0.2$ at $1/8$ doping.



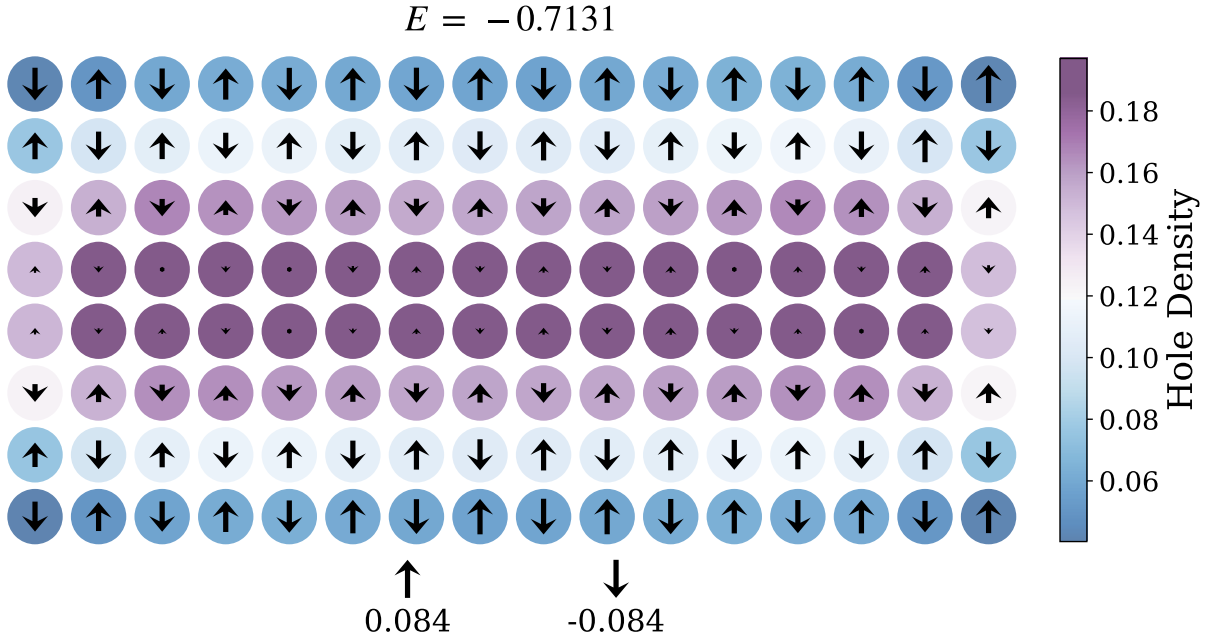
Supplementary Figure 11 | Hole and spin density distributions in the ground state for Hubbard with $U = 8$ and hole doping $\delta = 1/8$. The magnitudes of the spin density are represented by the sizes of the arrows while the direction is denoted by the direction of the arrows. Hole density is depicted using a color scale. The system is the pure Hubbard model with size 8×8 under OBC.



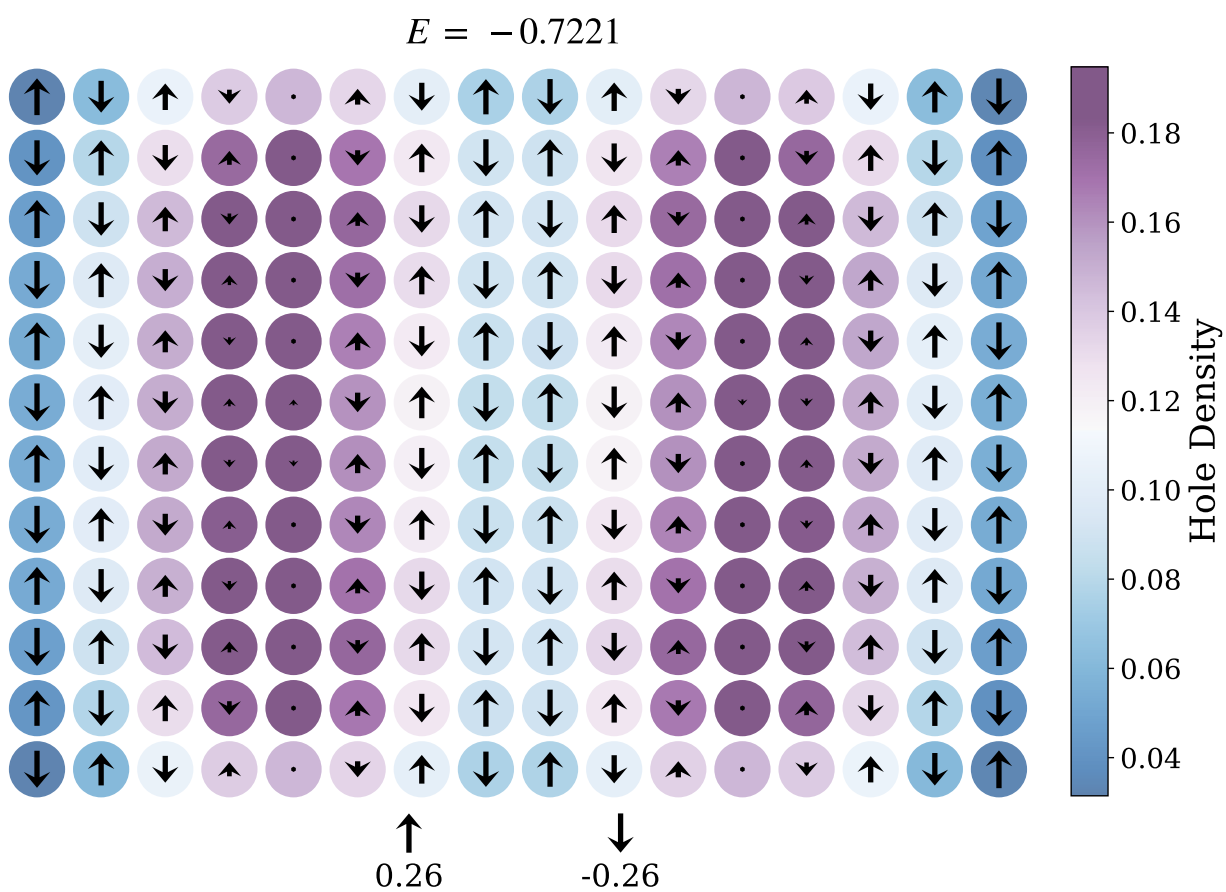
Supplementary Figure 12 | Similar as Figure 11 but for 16×4 pure Hubbard model under OBC.



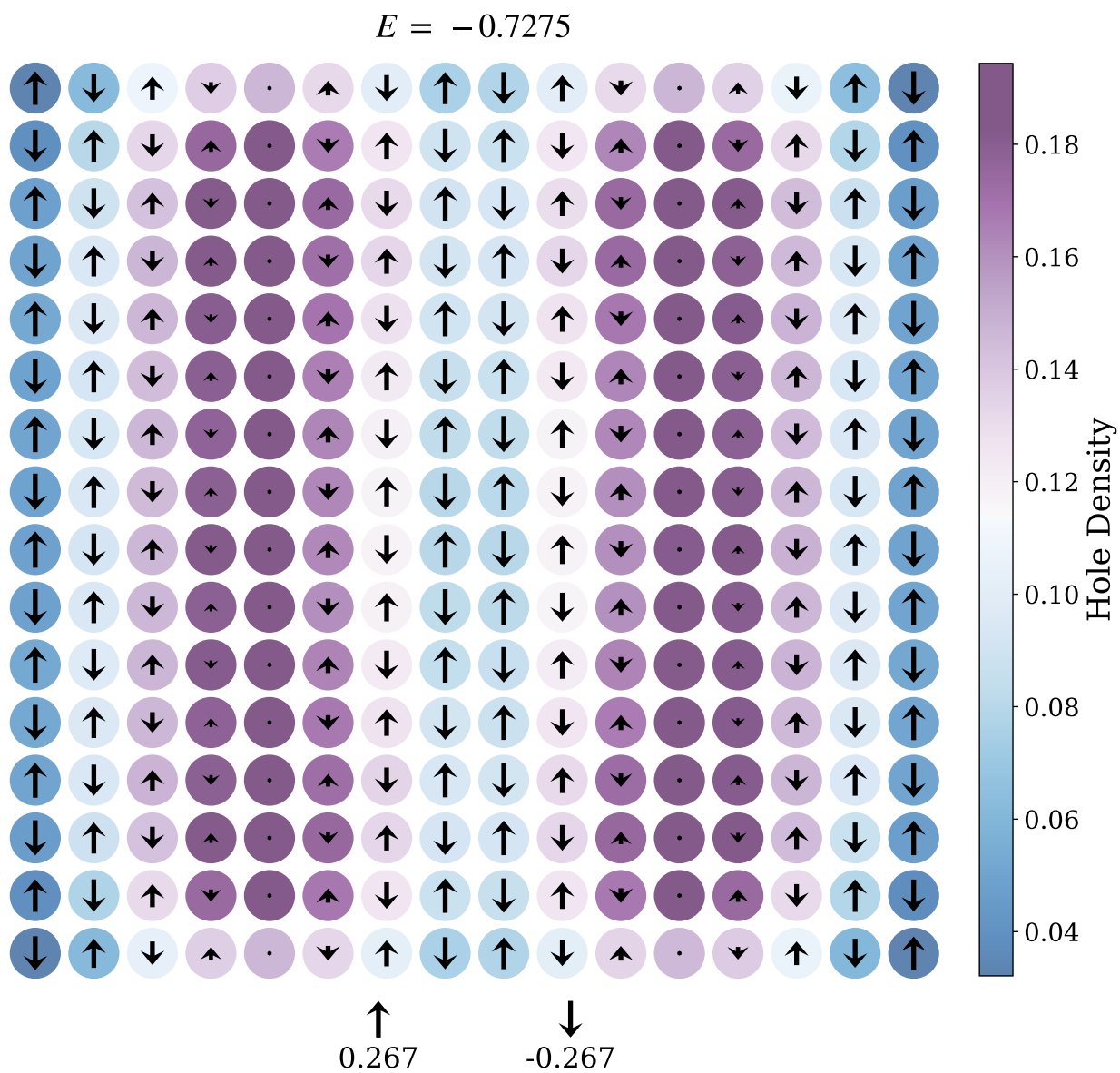
Supplementary Figure 13 | Similar as Figure 11 but for 16×6 pure Hubbard model under OBC.



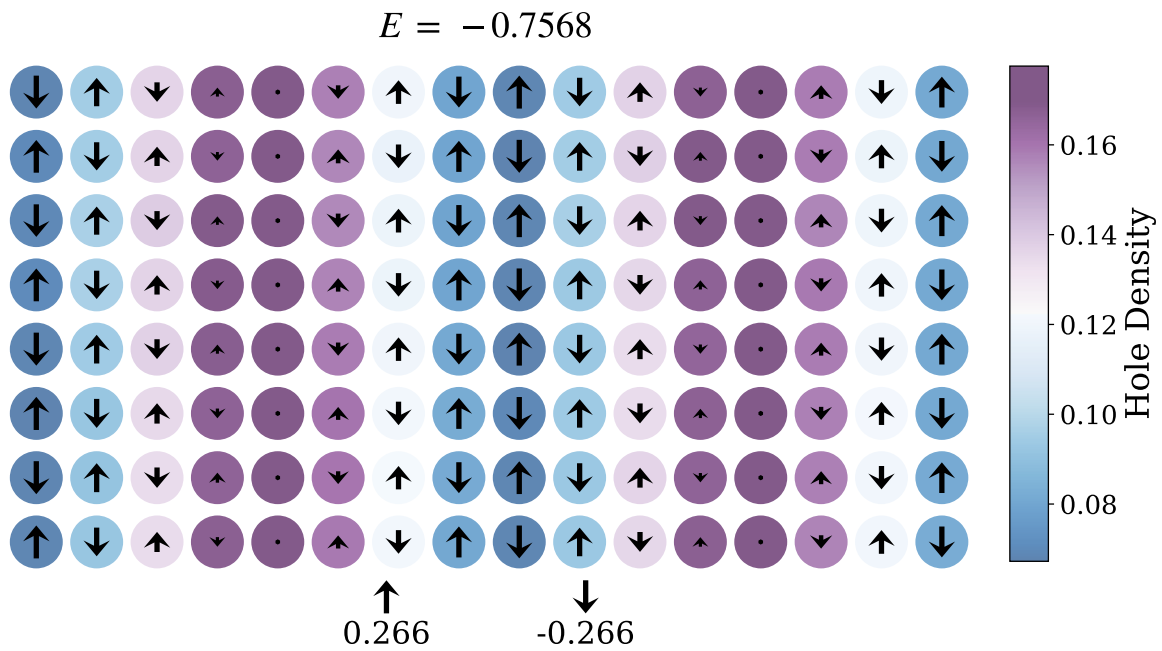
Supplementary Figure 14 | Similar as Figure 11 but for 16×8 pure Hubbard model under OBC.



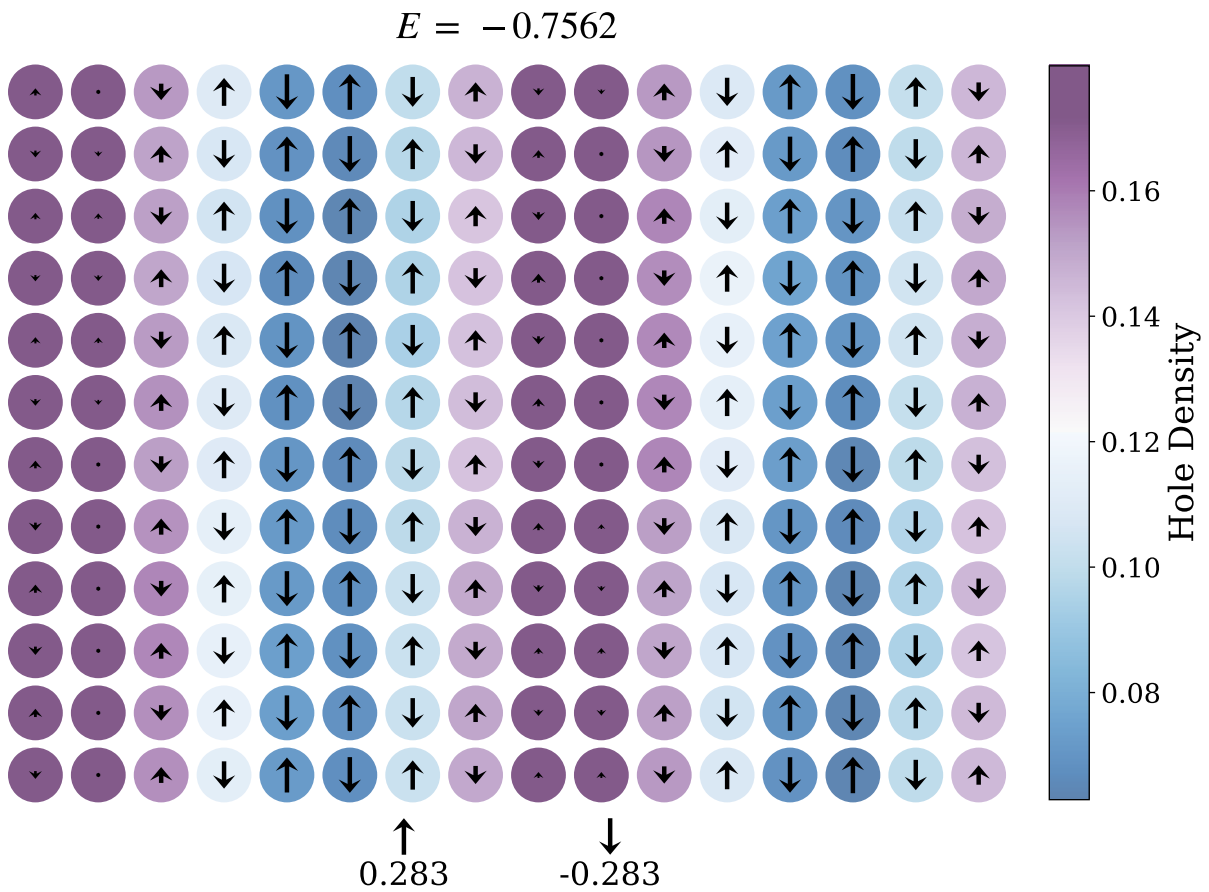
Supplementary Figure 15 | Similar as Figure 11 but for 16×12 pure Hubbard model under OBC.



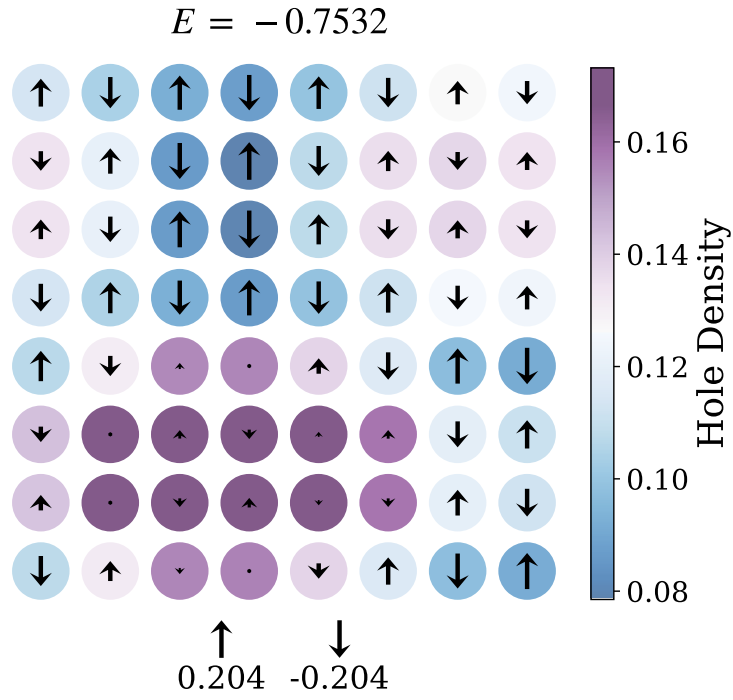
Supplementary Figure 16 | Similar as Figure 11 but for 16×16 pure Hubbard model under OBC.



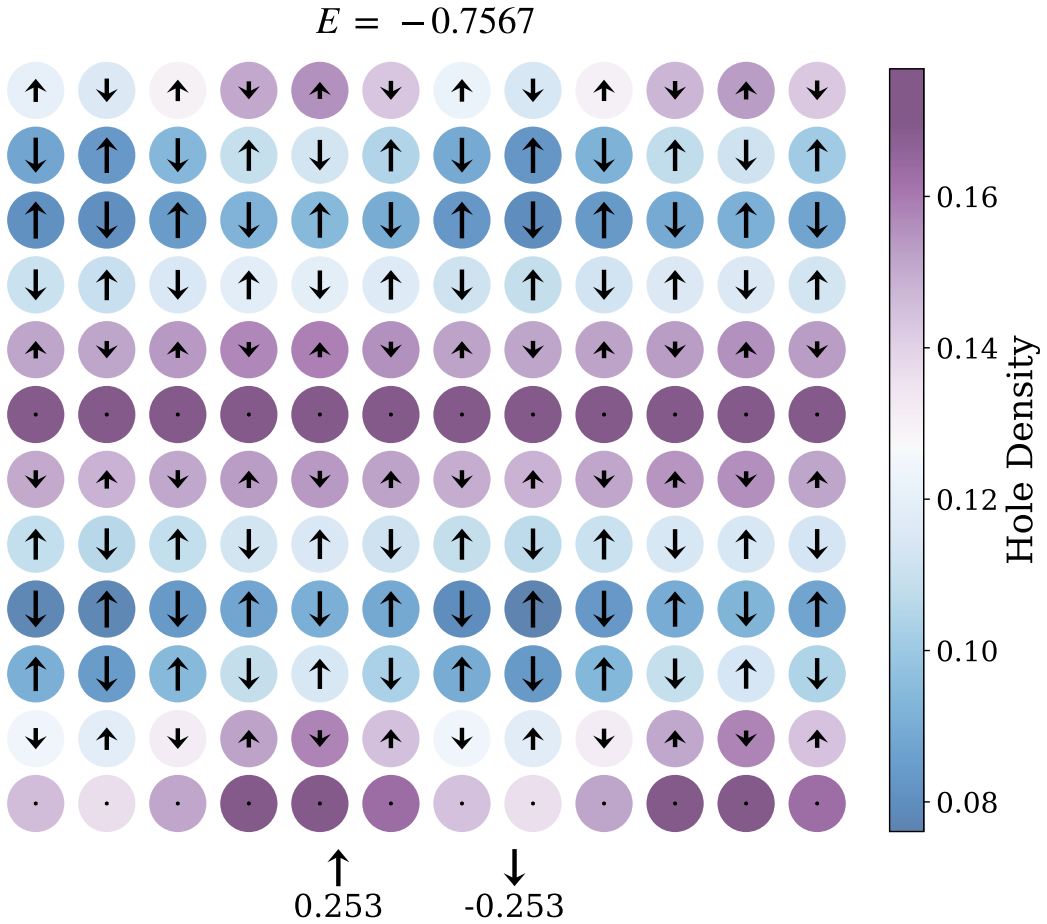
Supplementary Figure 17 | Similar as Figure 11 but for 16×8 pure Hubbard model under PBC.



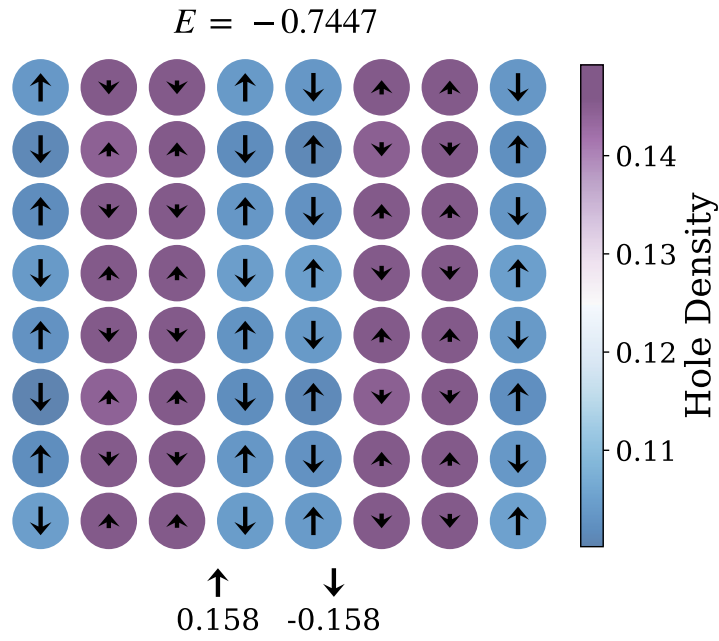
Supplementary Figure 18 | Similar as Figure 11 but for 16×12 pure Hubbard model under PBC.



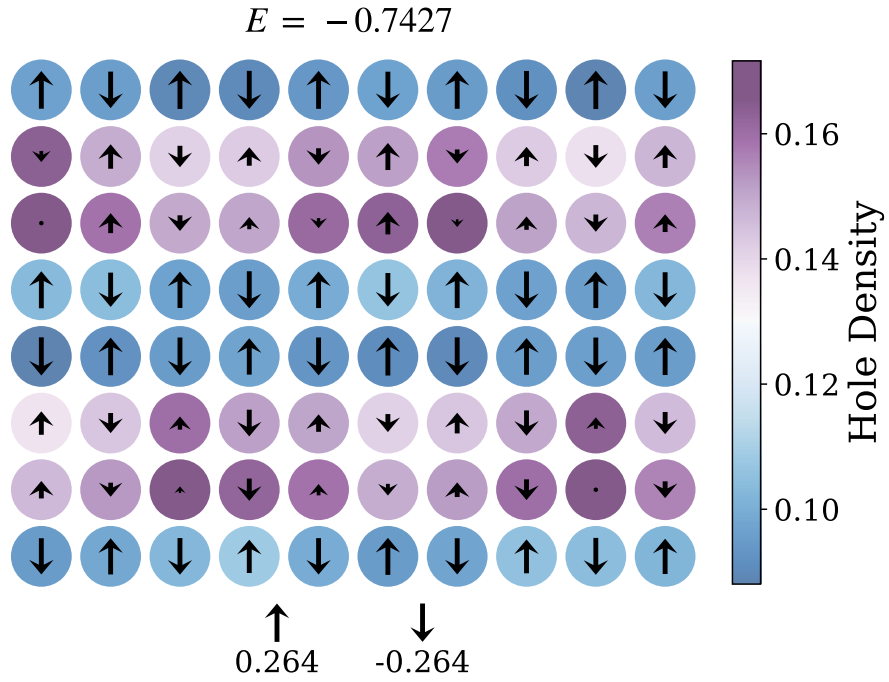
Supplementary Figure 19 | Similar as Figure 11 but for 8×8 pure Hubbard model under PBC.



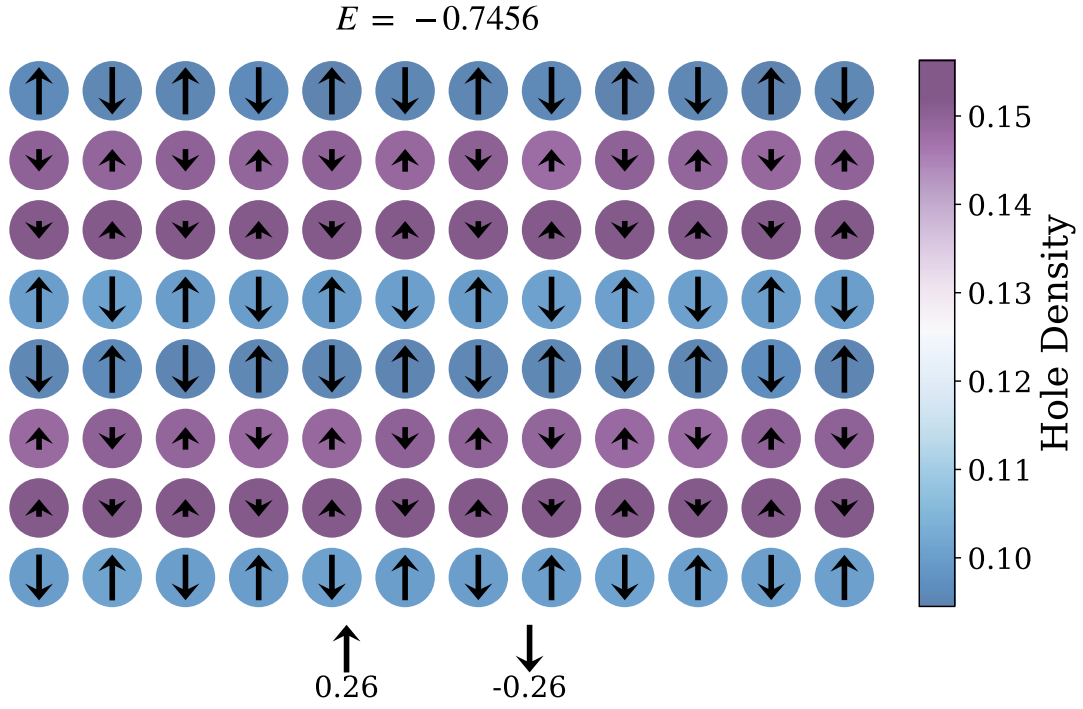
Supplementary Figure 20 | Similar as Figure 11 but for 12×12 pure Hubbard model under PBC.



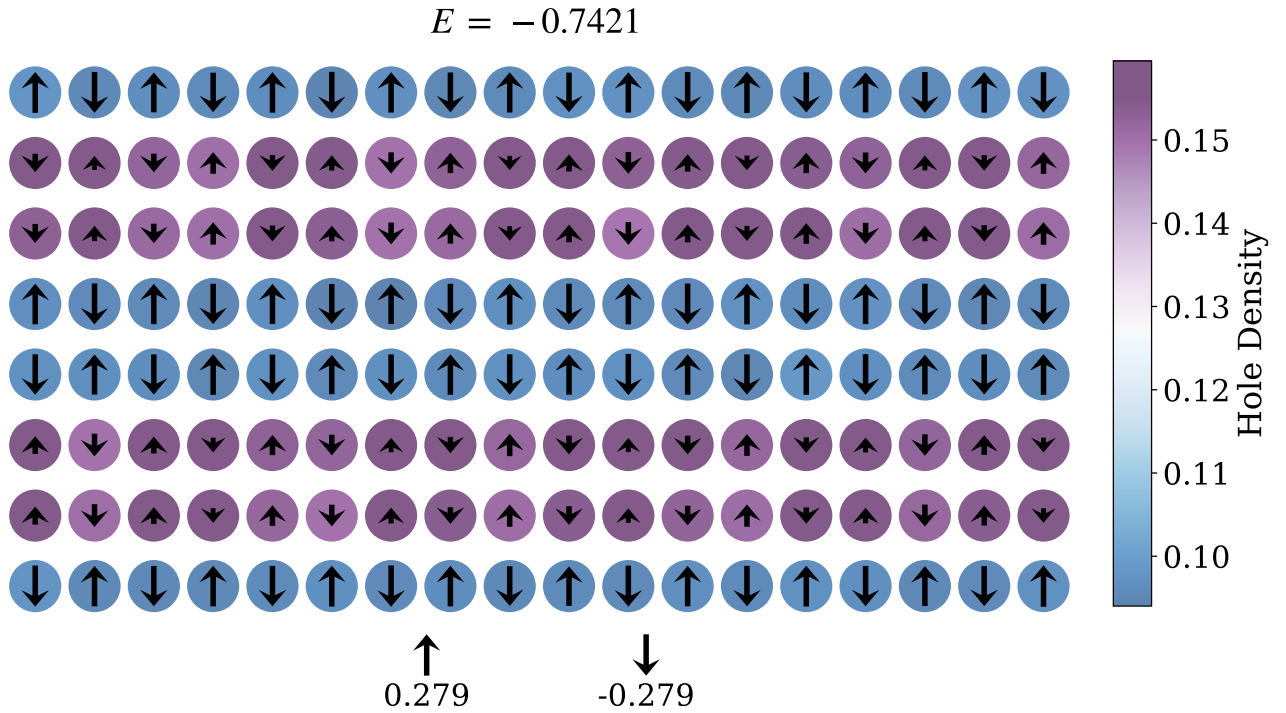
Supplementary Figure 21 | Similar as Figure 11 but for 8×8 Hubbard model with $t' = -0.2$ under PBC.



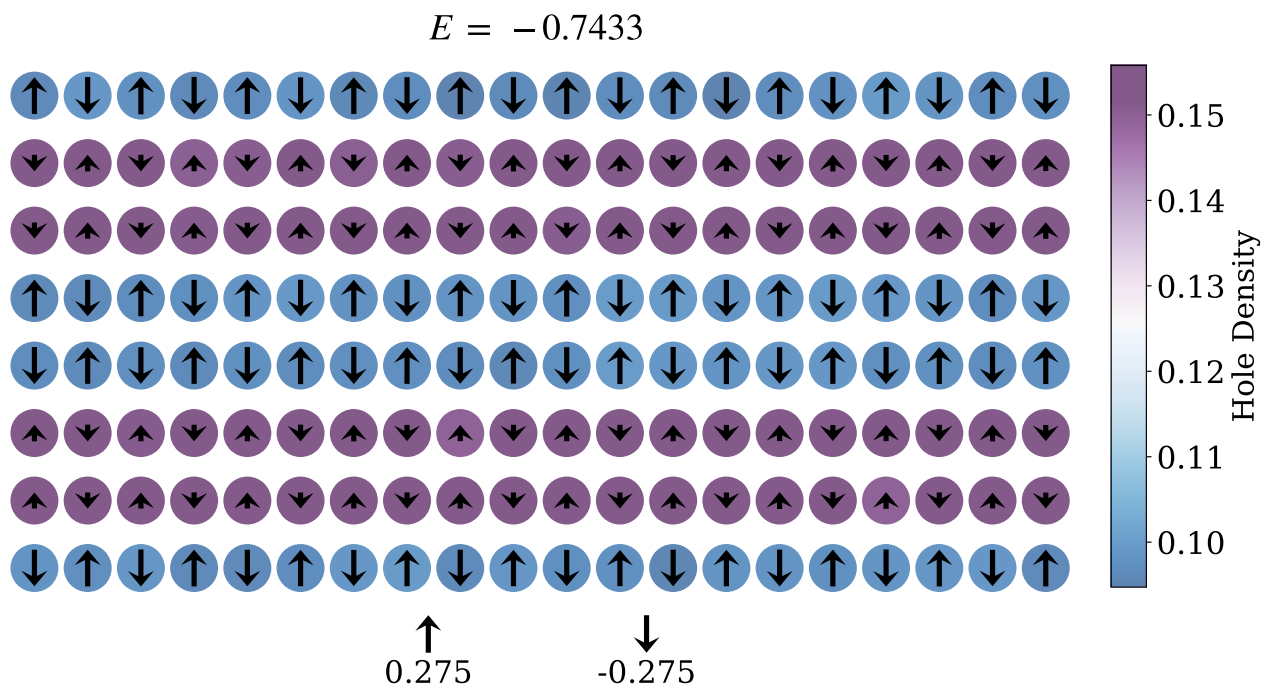
Supplementary Figure 22 | Similar as Figure 11 but for 10×8 Hubbard model with $t' = -0.2$ under PBC.



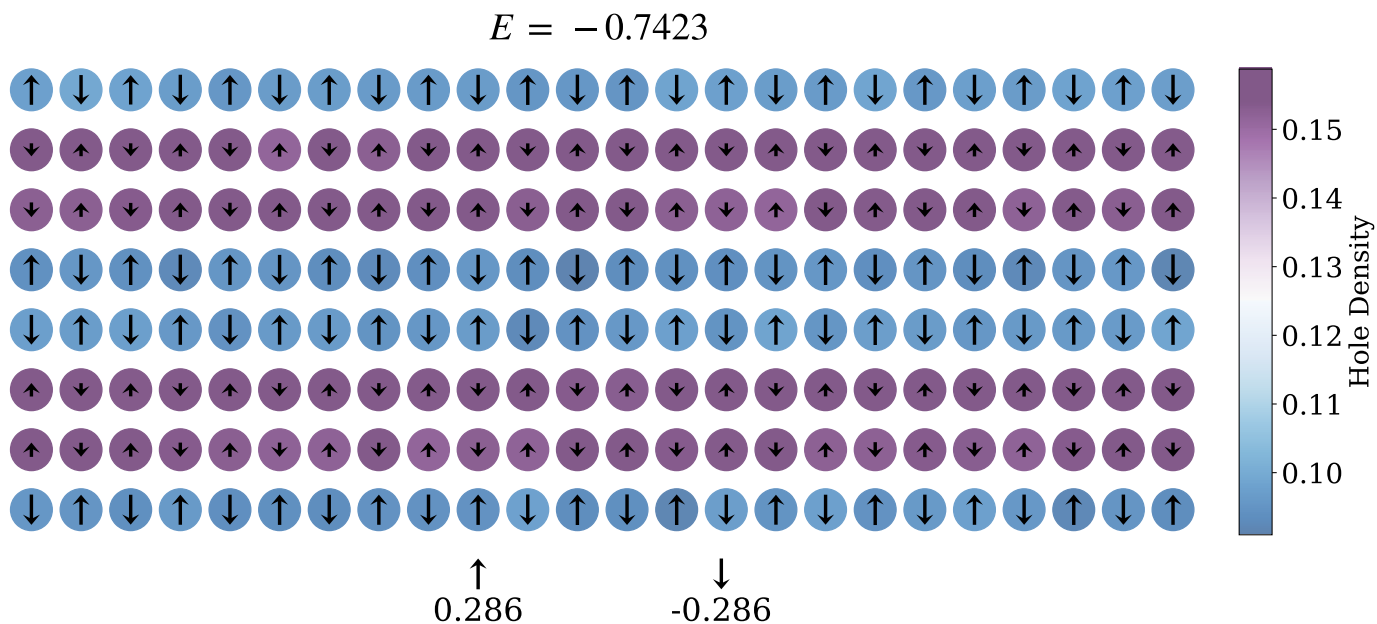
Supplementary Figure 23 | Similar as Figure ?? but for 12×8 Hubbard model with $t' = -0.2$ under PBC.



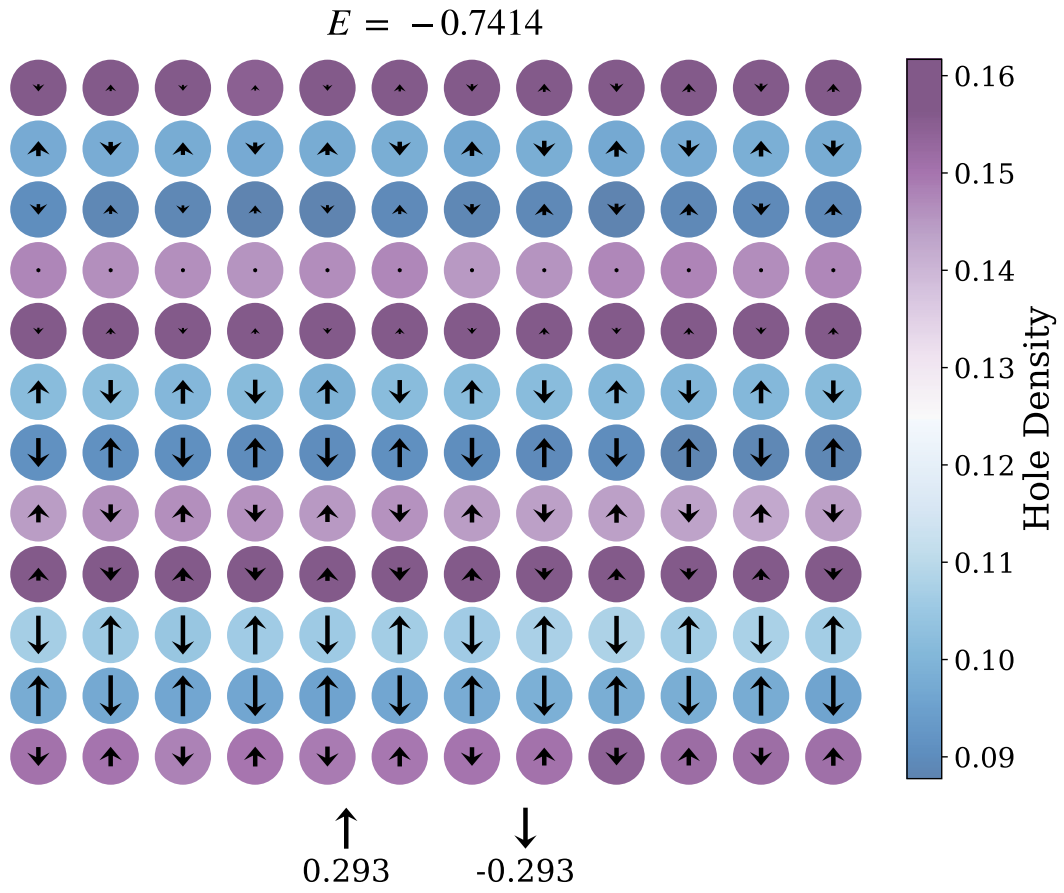
Supplementary Figure 24 | Similar as Figure 11 but for 18×8 Hubbard model with $t' = -0.2$ under PBC.



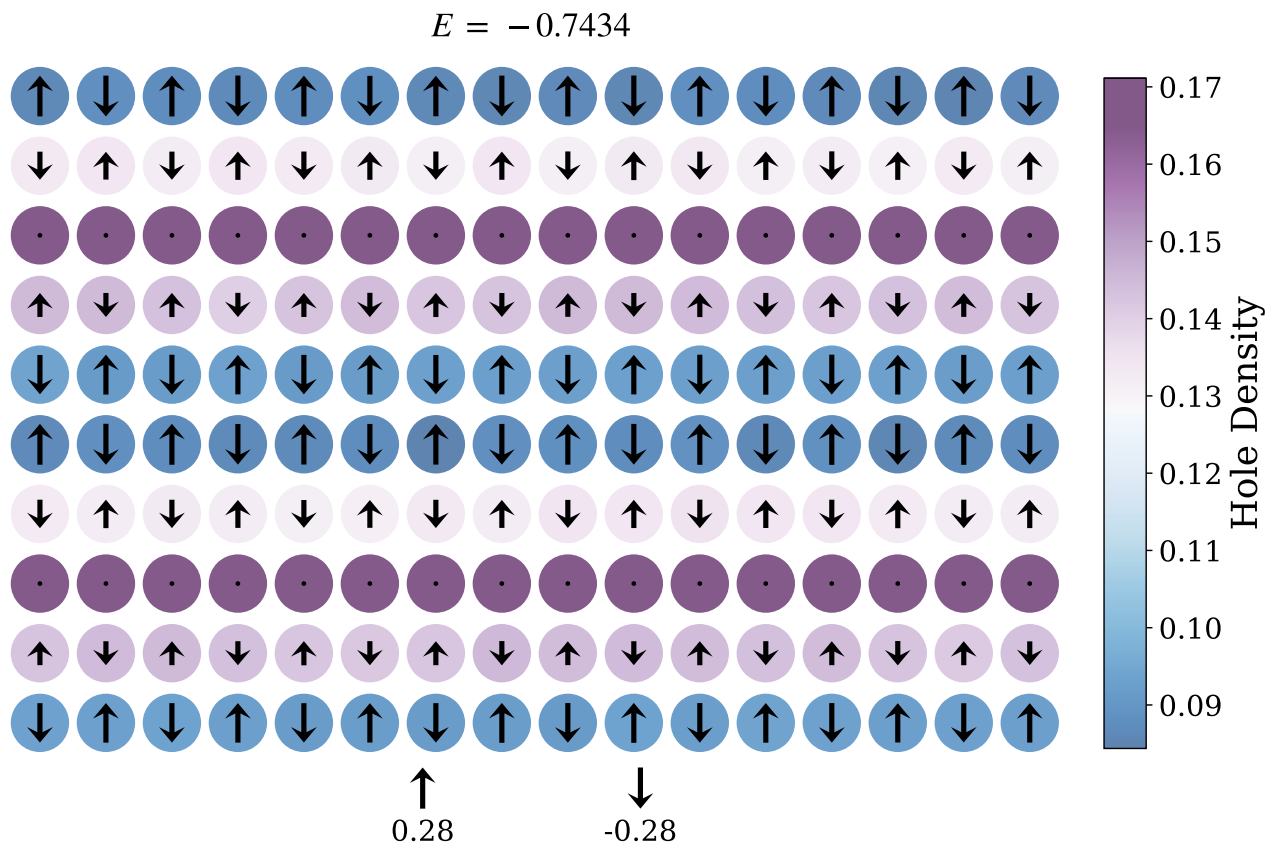
Supplementary Figure 25 | Similar as Figure 11 but for 20×8 Hubbard model with $t' = -0.2$ under PBC.



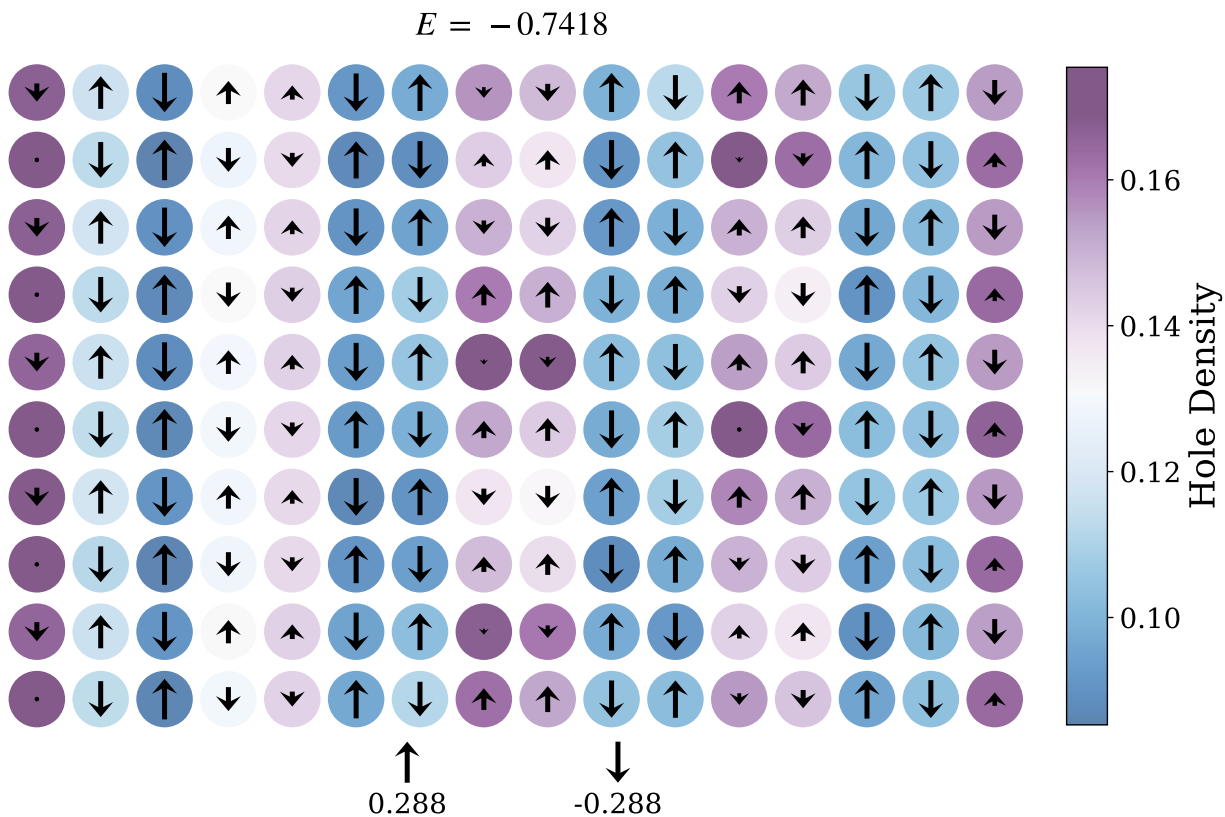
Supplementary Figure 26 | Similar as Figure 11 but for 24×8 Hubbard model with $t' = -0.2$ under PBC.



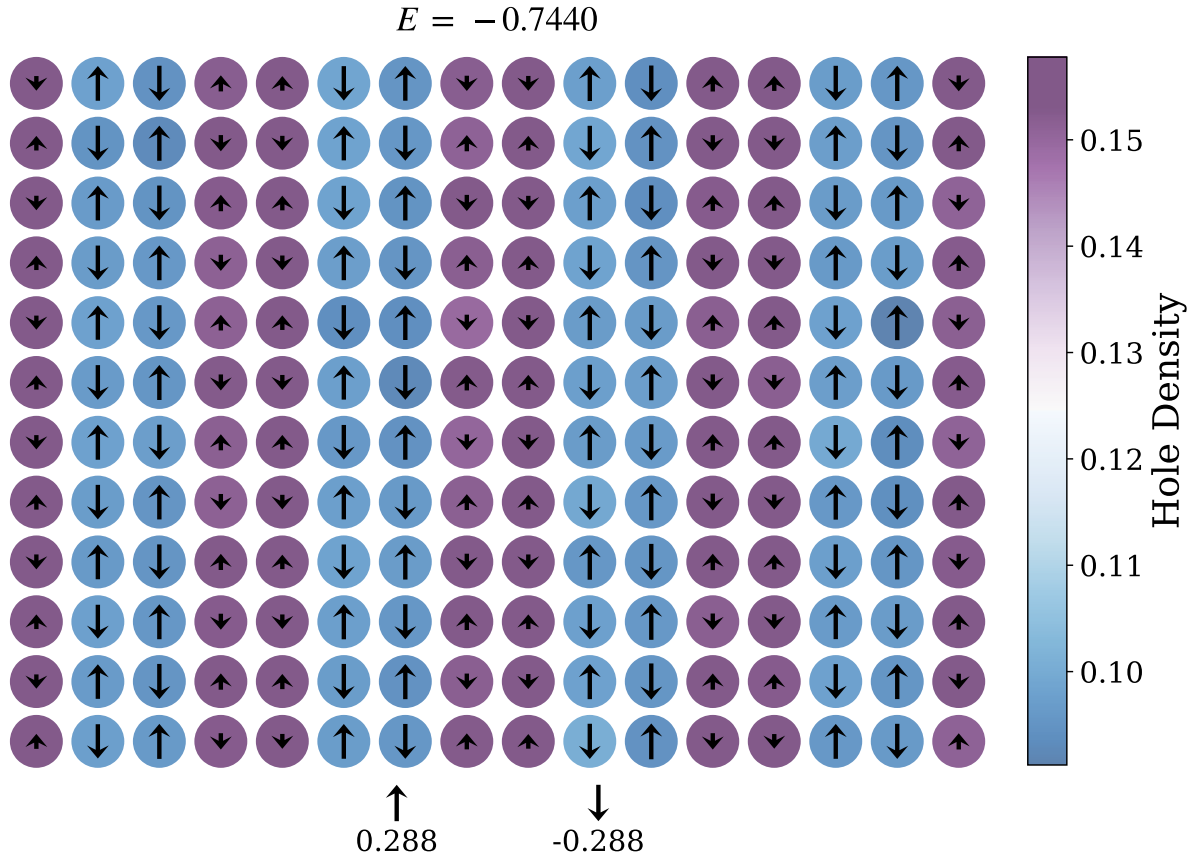
Supplementary Figure 27 | Similar as Figure 11 but for 12×12 Hubbard model with $t' = -0.2$ under PBC.



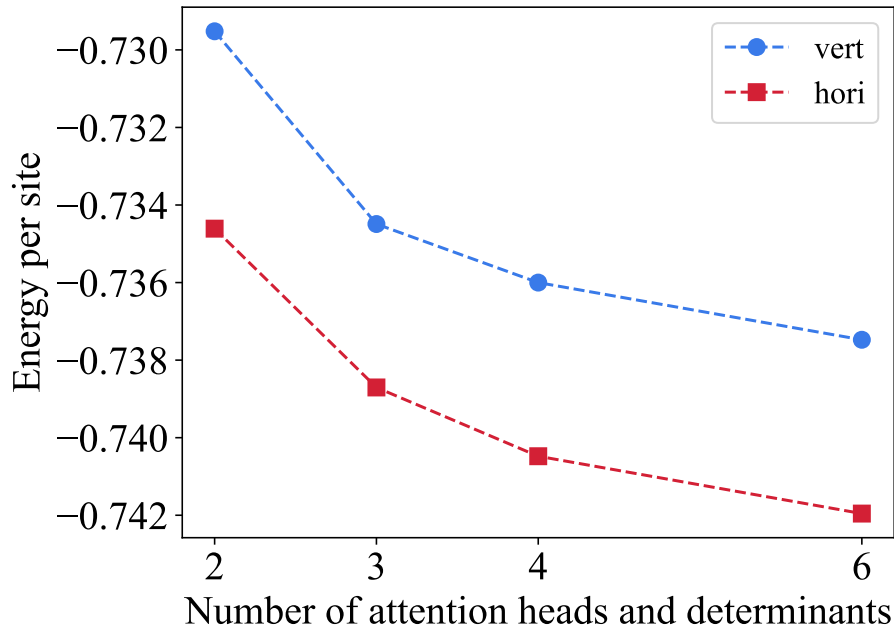
Supplementary Figure 28 | Similar as Figure 11 but for 16×10 Hubbard model with $r' = -0.2$ under PBC.



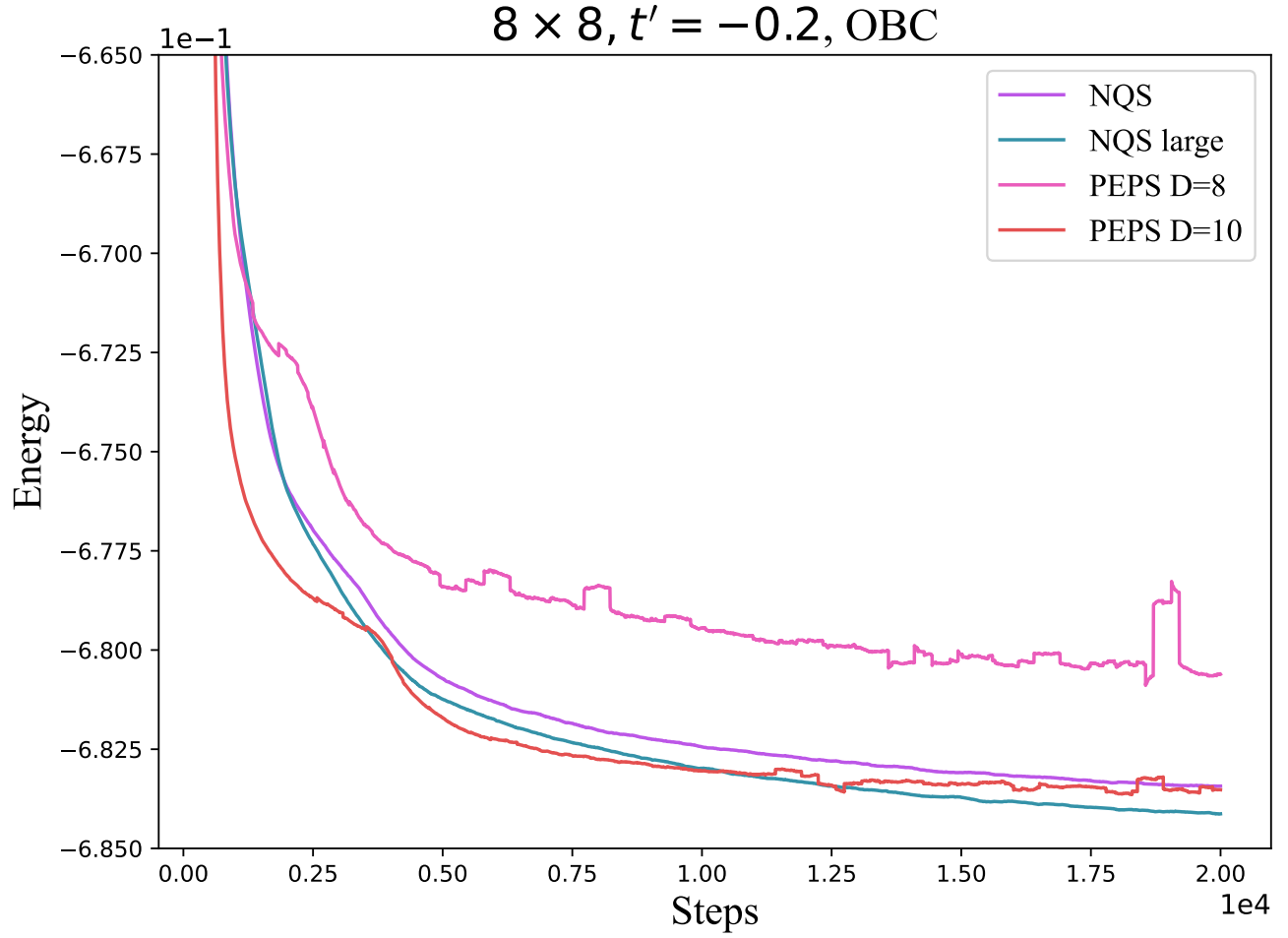
Supplementary Figure 29 | Similar as Figure 11 but for 16×10 Hubbard model with $r' = -0.2$ under PBC. We apply a temporary pinning field to get the vertical stripe.



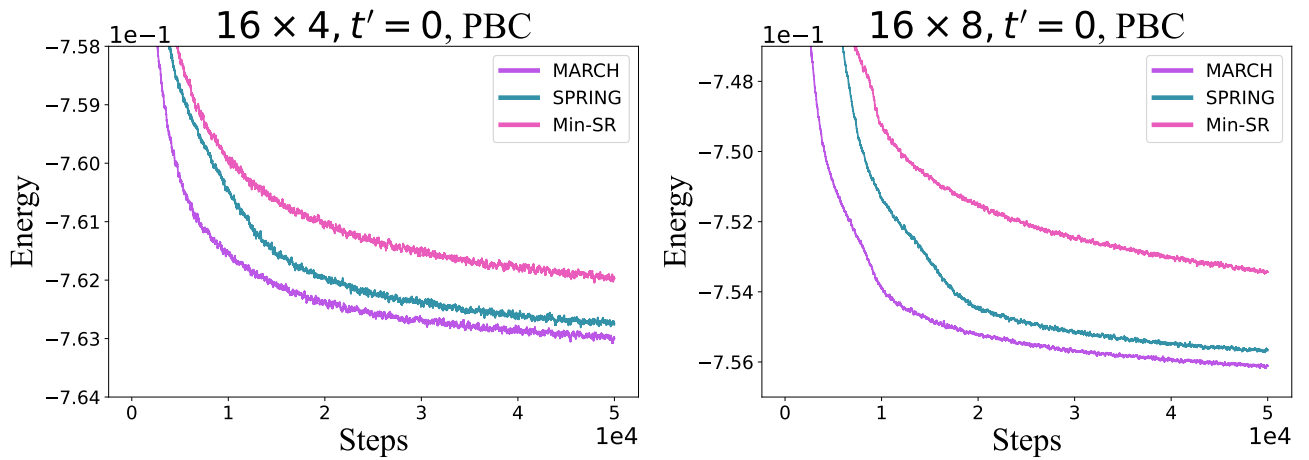
Supplementary Figure 30 | Similar as Figure 11 but for 16×12 Hubbard model with $t' = -0.2$ under PBC.



Supplementary Figure 31 | Vertical v.s. Horizontal stripe for the 32×8 Hubbard model with $t' = -0.2$ and $\delta = 1/8$, using NQS with up to 6 attention heads and determinants.



Supplementary Figure 32 | NQS v.s. PEPS under the same VMC batchsize. NQS denotes the hidden dimension of 256 and 4 determinants, NQS large denotes the hidden dimension of 384 and 6 determinants. D denotes the bond dimension in PEPS. The expressive power of NQS is similar to PEPS with $D = 10$.



Supplementary Figure 33 | Ablation study of optimizer. The results show that our optimizer converges significantly faster than SPRING and Min-SR.

Config	Value
Optimizer	MARCH
Optimizer momentum	$\mu, \beta = 0.95, 0.995$
Damping	$\lambda = 0.001$
Batch size	4096
Norm constraint at time t	$10^{-1}(1 + \max(t - 8000, 0)/8000)^{-1}$
Local energy clipping	5.0
MCMC step	$2.5L_x \times L_y$
Hidden dimension	256
Layers	4
Number of determinants	4
Steps	100000

Supplementary Table 7 | List of configs for the main training phase.

References

- [1] Di Luo and Bryan K Clark. Backflow transformations via neural networks for quantum many-body wave functions. *Physical review letters*, 122(22):226401, 2019.
- [2] Mingpu Qin, Hao Shi, and Shiwei Zhang. Benchmark study of the two-dimensional hubbard model with auxiliary-field quantum monte carlo method. *Physical Review B*, 94(8):085103, 2016.
- [3] Yusuke Nomura, Andrew S Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152, 2017.
- [4] Koji Inui, Yasuyuki Kato, and Yukitoshi Motome. Determinant-free fermionic wave function using feed-forward neural networks. *Physical Review Research*, 3(4):043126, 2021.
- [5] Javier Robledo Moreno, Giuseppe Carleo, Antoine Georges, and James Stokes. Fermionic wave functions from neural-network constrained hidden states. *Proceedings of the National Academy of Sciences*, 119(32):e2122059119, 2022.
- [6] Yu-Tong Zhou, Zheng-Wei Zhou, and Xiao Liang. Solving fermi-hubbard-type models by tensor representations of backflow corrections. *Physical Review B*, 109(24):245107, 2024.
- [7] Yantao Wu and Zhehao Dai. Algorithms for variational monte carlo calculations of fermion peps in the swap gates formulation, 2025.
- [8] Gil Goldshlager, Nilin Abrahamsen, and Lin Lin. A kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions. *Journal of Computational Physics*, 516:113351, 2024.
- [9] Ao Chen and Markus Heyl. Empowering deep neural quantum states through efficient optimization. *Nature Physics*, 20(9):1476–1481, 2024.
- [10] Riccardo Rende, Luciano Loris Viteritti, Lorenzo Bardone, Federico Becca, and Sebastian Goldt. A simple linear algebra identity to optimize large-scale neural network quantum states. *Communications Physics*, 7(1):260, 2024.