Supplemental Material: Electronic structure prediction of medium and high-entropy alloys across composition space

Shashank Pathrudkar,[1] Stephanie Taylor,[2] Abhishek Keripale,[1] Abhijeet Sadashiv Gangan,[2] Ponkrshnan Thiagarajan,[1] Shivang Agarwal,[3] Jaime Marian,[2] Susanta Ghosh,[1, 4, *] and Amartya S. Banerjee[2, †]

[1]*Department of Mechanical and Aerospace Engineering, Michigan Technological University*
[2]*Department of Materials Science and Engineering,*
University of California, Los Angeles, CA 90095, USA
[3]*Department of Electrical and Computer Engineering,*
University of California, Los Angeles, CA 90095, USA
[4]*Faculty member of the Center for Artificial Intelligence, Michigan Technological University*

(Dated: October 7, 2024)

A. Error Calculations

The primary metrics used for error calculations in this work are:

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{N_{\mathrm{grid}}}\sum_{i=1}^{N_{\mathrm{grid}}}(\rho_i - \hat{\rho}_i)^2}}{\frac{1}{N_{\mathrm{grid}}}\sum_{i=1}^{N_{\mathrm{grid}}}|\rho_i|}$$

$$\text{Relative } L_1 = \frac{\sum_{i=1}^{N_{\mathrm{grid}}}|\rho_i - \hat{\rho}_i|}{\sum_{i=1}^{N_{\mathrm{grid}}}|\rho_i|}$$

where, ρ is the ground truth electron density and $\hat{\rho}$ is the ML-predicted electron density.

The error in energy predictions presented in the paper are absolute difference in Kohn-Sham Density Functional Theory (KS-DFT) obtained energy and energy obtained by postprocessing ML predicted electron density field, and is reported in Hartree/atom.

B. Compositions Used in the *ab initio* Calculations

Training and testing compositions for SiGeSn are given in Figure S1, while for CrFeCoNi are given in Figure S2. The complete list of training and testing compositions for the 64 and 216 atom SiGeSn data are given in Tables S1 and S2, respectively. Finally, Table S3 contains the complete list of training and testing compositions in the 32 atom CrFeCoNi data.

System				System			
Index	% Si	% Ge	% Sn	Index	% Si	% Ge	% Sn
t-1	0	0	100	t-24	25	75	0
t-2	0	12.5	87.5	t-25	37.5	0	62.5
t-3	0	25	75	t-26	37.5	12.5	50
t-4	0	37.5	62.5	t-27	37.5	25	37.5
t-5	0	50	50	t-28	37.5	37.5	25
t-6	0	62.5	37.5	t-29	37.5	50	12.5
t-7	0	75	25	t-30	37.5	62.5	0
t-8	0	87.5	12.5	t-31	50	0	50
t-9	0	100	0	t-32	50	12.5	37.5
t-10	12.5	0	87.5	t-33	50	25	25
t-11	12.5	12.5	75	t-34	50	37.5	12.5
t-12	12.5	25	62.5	t-35	50	50	0
t-13	12.5	37.5	50	t-36	62.5	0	37.5
t-14	12.5	50	37.5	t-37	62.5	12.5	25
t-15	12.5	62.5	25	t-38	62.5	25	12.5
t-16	12.5	75	12.5	t-39	62.5	37.5	0
t-17	12.5	87.5	0	t-40	75	0	25
t-18	25	0	75	t-41	75	12.5	12.5
t-19	25	12.5	62.5	t-42	75	25	0
t-20	25	25	50	t-43	87.5	0	12.5
t-21	25	37.5	37.5	t-44	87.5	12.5	0
t-22	25	50	25	t-45	100	0	0
t-23	25	62.5	12.5				

Supplementary Table S1. **List of training and testing compositions in the 64 atom SiGeSn data set.** The error in electron density and energy prediction for these composition is given in S10 and Main Text Figure 4.

C. Details of Data Generation Methodology:

In the context of this work, the 'data' that was used for both testing and training the model consisted of a large batch of snapshots. Each snapshot represents an atomic arrangement in the simulation cell. For each of the snapshots, a grid in real space is considered and the corresponding electron density value at each grid point is obtained. The process of obtaining these 'snapshot' files was as follows:

1. An atomic configuration was selected.

[*] susantag@mtu.edu
[†] asbanerjee@ucla.edu

System Index	% Si	% Ge	% Sn
T-1	9.259	29.63	61.111
T-2	9.259	64.815	25.926
T-3	29.63	60.185	10.185
T-4	64.815	25	10.185
T-5	60.185	9.259	30.556
T-6	30.093	30.093	39.815
T-7	30.093	39.815	30.093
T-8	25	9.259	65.741
T-9	39.815	30.093	30.093
T-10	30.556	14.815	54.63
T-11	50	14.815	35.185
T-12	54.63	30.556	14.815
T-13	0	50	50
T-14	50	0	50
T-15	50	50	0
T-16	100	0	0
T-17	0	100	0
T-18	0	0	100
T-19	35.185	50	14.815
T-20	14.815	54.63	30.556
T-21	14.815	35.185	50

Supplementary Table S2. **List of compositions in the 216-atom SiGeSn data set. These systems are used for testing purposes only.** The error in electron density and energy prediction for these composition is given in Figure S4.

System Index	% Cr	% Fe	% Co	% Ni	System Index	% Cr	% Fe	% Co	% Ni
q-1	25	25	25	25	q-36	12.5	12.5	12.5	62.5
q-2	100	0	0	0	q-37	12.5	12.5	25	50
q-3	0	100	0	0	q-38	12.5	12.5	37.5	37.5
q-4	0	0	100	0	q-39	12.5	12.5	50	25
q-5	0	0	0	100	q-40	12.5	12.5	62.5	12.5
q-6	75	25	0	0	q-41	12.5	25	12.5	50
q-7	25	75	0	0	q-42	12.5	25	25	37.5
q-8	75	0	0	25	q-43	12.5	25	37.5	25
q-9	25	0	0	75	q-44	12.5	25	50	12.5
q-10	0	0	75	25	q-45	12.5	37.5	12.5	37.5
q-11	0	0	25	75	q-46	12.5	37.5	25	25
q-12	0	75	25	0	q-47	12.5	37.5	37.5	12.5
q-13	0	25	75	0	q-48	12.5	50	12.5	25
q-14	0	75	0	25	q-49	12.5	50	25	12.5
q-15	0	25	0	75	q-50	12.5	62.5	12.5	12.5
q-16	75	0	25	0	q-51	25	12.5	12.5	50
q-17	25	0	75	0	q-52	25	12.5	25	37.5
q-18	50	50	0	0	q-53	25	12.5	37.5	25
q-19	50	0	50	0	q-54	25	12.5	50	12.5
q-20	50	0	0	50	q-55	25	25	12.5	37.5
q-21	0	50	50	0	q-56	25	25	37.5	12.5
q-22	0	50	0	50	q-57	25	37.5	12.5	25
q-23	0	0	50	50	q-58	25	37.5	25	12.5
q-24	50	25	25	0	q-59	25	50	12.5	12.5
q-25	50	25	0	25	q-60	37.5	12.5	12.5	37.5
q-26	50	0	25	25	q-61	37.5	12.5	25	25
q-27	25	50	25	0	q-62	37.5	12.5	37.5	12.5
q-28	25	50	0	25	q-63	37.5	25	12.5	25
q-29	0	50	25	25	q-64	37.5	25	25	12.5
q-30	25	25	50	0	q-65	37.5	37.5	12.5	12.5
q-31	25	0	50	25	q-66	50	12.5	12.5	25
q-32	0	25	50	25	q-67	50	12.5	25	12.5
q-33	25	25	0	50	q-68	50	25	12.5	12.5
q-34	25	0	25	50	q-69	62.5	12.5	12.5	12.5
q-35	0	25	25	50					

Supplementary Table S3. **List of training and testing compositions in the 32-atom CrFeCoNi data set**. Note that not all compositions are used for training. The error in electron density and energy prediction for these composition is given in Figures S11.

2. A Kohn-Sham density functional theory (KS-DFT) calculation was performed to obtain the ground state electron densities associated with that atomic configuration.

3. Text-processing was performed on the electron density output to format it for further calculations.

For each input atomic configuration, one snapshot file would be obtained, which could then be used for either training or testing.

To produce the atomic configurations, one of three options was leveraged: sampling from the trajectory of an *ab initio* molecular dynamics (AIMD) simulation, sampling from the trajectory of a classical molecular dynamics (MD) simulation, or handcrafting more unique systems for the purpose of test data (e.g. defects, checkerboard boundaries).

The first option, AIMD, was used for all of the SiGeSn system data (ternary, binary, and unary derivatives). This option was selected due to the ease of implementation; we obtained coordinates and electron densities, and both of these were calculated with high fidelity at each step of the AIMD trajectory. We leveraged this approach in our previous work [1], and it was straightforward to extend the method to the ternary SiGeSn system. This streamlined many aspects of the data generation. However, the downside of this approach is its high computational expense; since most of the steps in the trajectory are not included in the final data pool (they are simply intermediate steps), a great deal of computational resources go into electronic structure calculations that are — in the context of data generation — unused.

In an effort to leverage a significantly cheaper alternative to AIMD, the second option, classical molecular dynamics (MD), was used generating the atomic configurations for all of the CrFeCoNi system data (quaternary, ternary, binary, and unary derivatives). Since a quaternary system has a higher degree of freedom with respect to the compositions, a greater number of simulations were required to obtain an adequate pool of data across composition space. Furthermore, each electronic structure calculation was itself more expensive, due to the inclusion of semi-core electrons for the atomic species Cr, Fe, Co, and Ni. Leveraging MD meant introducing an additional step into the data generation pipeline, but it vastly increased the efficiency at which atomic configurations and their corresponding electron densities could be obtained.

Lattice constant calculations: One parameter needed as input in both our MatGL-enabled MD simulations and SPARC-enabled AIMD and static KS-DFT calculations is the lattice constant. For a flexible approach that allowed for the simulation of any composition choice

Element	Lattice Geometry	Lattice Constant Obtained	Materials Project Reference	Experimental Reference [3]
Si	diamond	**5.47**	*5.44*	*5.42*
Ge	diamond	**5.76**	*5.67*	*5.65*
Sn	diamond	**6.63**	*6.57*	*6.46*
Cr	FCC	**3.65**	*3.58*	-
Cr	BCC	*2.86*	*2.97*	*2.88*
Fe	FCC	**3.45**	*3.66*	*3.56*
Fe	BCC	*2.76*	*2.86*	*2.93*
Co	FCC	**3.45**	*3.51*	*3.55*
Ni	FCC	**3.51**	*3.48*	*3.52*

Supplementary Table S4. Lattice constants obtained from the pseudopotentials employed in this study, in units of Angstrom, for the 8-atom diamond unit cell, the 4-atom face-centered cubic (FCC) unit cell, and the 2-atom body-centered cubic (BCC) unit cell. Values in **bold** were leveraged for the rule-of-mixtures appraoch. Values in *italics* were not leveraged, but are provided here to for the purpose of comparison. Note that while Fe and Co both exhibit FCC phases at higher temperature, Cr does not have an experimentally observed FCC phase in nature. The lattice constant for FCC Cr was used to match the FCC lattice imposed upon our CrFeCoNi alloy systems.

Alloy Composition	M3GNet-Optimized Lattice Constant	Vegard's Law Lattice Constant
Cr	3.61	3.65
Fe	3.46	3.45
Ni	3.50	3.51
Co	3.52	3.45
$Cr_{0.5}Fe_{0.5}$	3.57	3.55
$Fe_{0.5}Ni_{0.5}$	3.56	3.48
$Ni_{0.5}Co_{0.5}$	3.50	3.48
$Cr_{0.5}Ni_{0.5}$	3.53	3.58
$Cr_{0.25}Fe_{0.5}Ni_{0.25}$	3.55	3.51
$Cr_{0.5}Fe_{0.25}Co_{0.25}$	3.55	3.55
$Cr_{0.5}Co_{0.25}Ni_{0.25}$	3.54	3.57
$Fe_{0.25}Co_{0.25}Ni_{0.5}$	3.53	3.51
$Cr_{0.25}Fe_{0.25}Co_{0.25}Ni_{0.25}$	3.54	3.52
$Cr_{0.125}Fe_{0.25}Co_{0.375}Ni_{0.25}$	3.53	3.49
$Cr_{0.25}Fe_{0.125}Co_{0.25}Ni_{0.375}$	3.52	3.52
$Cr_{0.25}Fe_{0.375}Co_{0.25}Ni_{0.125}$	3.55	3.51

Supplementary Table S5. Comparison between unit cell lattice parameter obtained from volume relaxation with MatGL and the unit cell lattice parameter obtained from the rule-of-mixtures weighted average approach described in the text. These MatGL simulations were done with 6x6x6 supercells of 864 atoms in FCC lattice geometry, and the lattice constants shown are scaled down by a factor of six to allow for facile comparison with Table S4. Units in Angstrom.

within our alloy systems, we opted for implementing the rule-of-mixtures, Vegard's Law [2]. First, we obtained the lattice constant predicted for the pure elements Si, Ge, Sn, Cr, Fe, Co, and Ni by each respective pseudopotential, imposing the specified lattice geometry. Due to their impact on the rest of the data generation, these calculations were done at a higher level of precision. Then, the lattice constant for any alloy composition was obtained by taking a weighted average of the lattice constants from the pure elements, in proportion to the elemental composition of that alloy. The pure lattice constants are shown in Table S4. While this is a simplifying assumption, it was sufficient for our purposes. It is encouraging to note that when the lattice parameter is relaxed with MatGL, the optimized lattice parameter obtained is not significantly different, as shown in Table S5. As a consequence the averaged electron density also approximately follows Vegard's law. This makes sense because the average electron density is contingent on the volume, and the volume of our cubic systems is simply the lattice constant cubed.

Data generation of SiGeSn systems: For SiGeSn, the fractional coordinates for an 8-atom unit cell of the diamond lattice structure were scaled up to produce 64-atom and 216-atom supercells. Atom labels were randomly appended to these coordinates, in accordance with the desired alloy composition. These initial atomic configurations were then converted to the SPARC-required format of a .ion file, and the fractional coordinates were scaled by the lattice constant obtained via Vegard's Law. For each .ion file, a corresponding .inpt file containing the AIMD settings for SPARC was generated. These files were then fed into SPARC and allowed to run, generating an atomic trajectory. After a short equilibration period to allow the temperature and energy fluctuations to stabilize, snapshots were extracted at fixed intervals and the corresponding electron densities associated with those snapshots were collected. For the 64-atom case, there were 45 unique compositions, 4 random initial atomic configurations for each composition, 1 AIMD temperature (2400K), and 6 snapshots collected from each AIMD run. This yielded a total of 900 data points. For the 216-atom case, there were 21 unique compositions, 1 random initial atomic configuration, 1 AIMD temperature (2400K), and 3 snapshots collected. This yielded a total of 63 data points. The 216-atom SiGeSn data was used for testing, while the 64-atom data was used for both training and testing. See Figure S1 (and **Tables S1** and **S2**) for the SiGeSn compositions.

Data generation of CrFeCoNi systems: For CrFeCoNi, the fractional coordinates for a 4-atom unit cell of the face-centered cubic (FCC) structure were scaled up to produce 32-atom supercells. Randomly selected atom labels were assigned to the coordinates, in proportion to the alloy composition. These initial atomic configurations were converted to a data structure that MatGL could read, a Python file containing the molecular dynamics settings was generated, and the molecular dynamics (MD) simulation was run. Snapshots were collected from the trajectory at fixed intervals after a short equilibration time. SPARC was then run, thus producing the electron densities associated with the extracted snapshots. For the 32-atom case, there were 69 unique compositions, 4 random initial atomic configurations per composition, 4 MD temperatures (4000K, 5000K, 6000K, 7000K), and 7 snapshots collected. This yielded a total of 7728 data points. The 32-atom data was used for both training and testing. See Figure S2 (and **Tables S3** for the CrFeCoNi compositions.

Data generation of special SiGeSn systems: Three forms of additional SiGeSn test data were produced for model generalizability studies: monovacancy-containing, divacancy-containing, and handcrafted 'checkerboard' systems (with species segregation). For convenience, the *atomsk* package [4] was used in generating some of these system configurations. For the monovacancy and divacancy data, atom labels were randomly assigned to the coordinate sites of a pristine 64-atom diamond cubic lattice cell such that the final composition matched one of twelve pre-selected compositions. To generate a monovacancy-containing system, a randomly selected atom that matched the desired atom type would be removed from the cell. By choosing to remove either a Si, Ge, or Sn atom, each snapshot produced three derivative data points. To generate a divacancy-containing system, a randomly selected atom, and its neighbor would be removed as a pair, producing one of six possible scenarios: a missing SiSi, GeGe, SnSn, SiGe, GeSn, or SiSn pair. Since it was also of interest to see how well the trained model could make predictions for intersections of bulk elemental regions, handcrafted 'checkerboard' systems were produced. Cubic simulation cells of 64 and 216 atoms occupying diamond lattice sites were divided up into smaller cubic sub-regions, i.e. either 8 bins (2x2x2) for the 64-atom and 216-atom cells, or 27 bins (3x3x3) for the 216-atom cell. Elemental labels were then assigned to each bin, such that no two neighboring bins contained the atoms of the same element, with periodic boundaries taken into consideration as well. In the 8-bin case, three compositions were considered: $Si_{0.25}Ge_{0.375}Sn_{0.375}$, $Si_{0.375}Ge_{0.25}Sn_{0.375}$, and $Si_{0.375}Ge_{0.375}Sn_{0.25}$. In the 27-bin case, just the equiatomic SiGeSn case was considered (e.g. $Si_{0.33}Ge_{0.33}Sn_{0.33}$).

In total, the data generated for this study consisted of:

- 1080 snapshots for the 64-atom SiGeSn system
- 63 snapshots for the 216-atom SiGeSn system
- 7728 snapshots for the 32-atom CrFeCoNi system
- 36 snapshots for the vacancy-containing SiGeSn systems
- 72 snapshots for the divacancy-containing SiGeSn systems
- 7 snapshots for the handcrafted, 'checkerboard' SiGeSn systems.

It is noted that being able to obtain more data more quickly with the augmented MD method enhances the feasibility of further studies with large system sizes and greater number of alloying elements.

D. Additional results

Additional supporting plots are presented in this section. Figure 8 b in the main text shows a plot of error in energy for the AL2 model trained on $\delta\rho$; for the sake of comparison, Figure S3 shown here presents error in energy for the AL2 model trained on ρ.

Figure S4 shows density and energy prediction errors for the larger 216-atom SiGeSn system; recall that only the 64-atom SiGeSn data was used in the training set.

In Figure 4 in the main text, SiGeSn prediction results for the Bayesian Active Learning approach are shown. Figure S5 shows the analogous results for the Tessellation approach. Also, Figure 4 in the main text has the same colorbar scales for AL1 and AL2 model. This is to illustrate the decrease in errors obtained from the AL2 model vs. the AL1 model. However, presenting the results in that fashion inhibits the readability of the AL2 values. To circumvent the readability issue, Figure S6 is presented below, with separate colorbar scales.

Figures S7, S8 and S9 demonstrate the difference between 'ground-truth' KS-DFT-obtained electron densities and ML-obtained electron densities for a sample of systems. Specifically, Figure S7 shows a SiGeSn vacancy-containing snapshot and Figure S8 shows a thermalized 216-atom SiGeSn snapshot and a thermalized 32-atom CrFeCoNi snapshot.

Figure 9 in the main text shows key results for the systems considered in this work, but the results shown there are aggregated over all compositions. Since the composition-dependent values may also be of interest, Figures S10 and S11 plot the error in electron density and error in energy by composition. Figure S10 shows the results for the 45 compositions present in the 64-atom SiGeSn test data. Figure S11 shows the results for the 69 compositions present in the 32-atom CrFeCoNi test data. Note that the SiGeSn results in Figure S10 come from a model trained on ρ (the charge density field), while the CrFeCoNi results in Figure S11 come from a model trained on $\delta\rho$ (the difference between the charge density field and atomic densities). In both cases, though, the density errors displayed are for the charge density itself.

E. Extension to Quinary System: AlCrFeCoNi

To further illustrate that our methodology works well for typical high entropy alloys, we trained an additional model based on the quinary AlCrFeCoNi system, and focused on near-equiatomic compositions.

The details of this model are as follows. Data was generated in the same fashion as the quaternary CrFeCoNi system; a crystalline 32-atom face-centered cubic supercell was assigned atom labels corresponding to different composition percentages. The twenty near-equi-atomic compositions that were selected for data generation are shown in Table S6. The crystalline system was set as the initial configuration for a classical molecular dynamics simulation performed at 4000K with MatGL [5, 6]. After an equilibration period of 1000 timesteps (each timestep is one femtosecond), snapshots were extracted from the trajectory, at 100-timestep intervals. Data generation was minimal; only one trajectory per alloy composition was employed, yielding a total of 10 configuration snapshots for each composition (the initial crystalline configuration and nine thermalized configurations drawn from the molecular dynamics trajectory). Twenty compositions with ten configurations each yielded a total data pool of 200 snapshots for this model.

We constrained our quinary AlCrFeCoNi model to the face-centered cubic (FCC) system only, neglecting any potential phase transitions to other lattices. The possibility of exploration of the utility of our ML model across crystal systems is left as future work.

The architecture of the AlCrFeCoNi neural network was identical to the other models produced in this study. The model was trained on data from 16 of the compositions, and the remaining 4 compositions were leveraged for testing. Since the $\rho-$SAD approach worked well at reducing the error metrics in the quaternary system, we trained the AlCrFeCoNi model using $\rho-$SAD instead of just ρ. Model training took only 14 CPU-hours for this system. Figure 9 in the main text shows the error in electron density and energy prediction, as obtained from the 4 near-equiatomic test compositions. The values obtained are just as good in accuracy as those obtained for the quaternary system, suggesting the successful application of the ML model to typical 5-element high entropy alloy systems.

System Index	% Al	% Cr	% Fe	% Co	% Ni
R-1	0	0	0	0	100
R-2	0	0	0	100	0
R-3	0	0	100	0	0
R-4	0	100	0	0	0
R-5	12.5	12.5	25	25	25
R-6	12.5	25	12.5	25	25
R-7	12.5	25	25	12.5	25
R-8	12.5	25	25	25	12.5
R-9	18.75	18.75	18.75	18.75	25
R-10	18.75	18.75	18.75	25	18.75
R-11	18.75	18.75	25	18.75	18.75
R-12	18.75	25	18.75	18.75	18.75
R-13	25	12.5	12.5	25	25
R-14	25	12.5	25	12.5	25
R-15	25	12.5	25	25	12.5
R-16	25	18.75	18.75	18.75	18.75
R-17	25	25	12.5	12.5	25
R-18	25	25	12.5	25	12.5
R-19	25	25	25	12.5	12.5
R-20	100	0	0	0	0

Supplementary Table S6. **List of compositions in the 32-atom AlCrFeCoNi data set.** Compositions R-8, R-9, R-10, and R-11 were used for testing and the other compositions were included in the training dataset. The four testing compositions were selected at random.

F. Efficiency Comparison

In the following, we elaborate on how much faster the ML approach is, for compositional space exploration compared to the conventional Kohn-Sham Density Functional Theory (KS-DFT) approach. A key feature of the analysis presented below is the inclusion of "offline" or training data generation costs, that is often left out in other similar studies [7]. We consider the following four systems presented in the paper:

- 64-atom SiGeSn system
- 216-atom SiGeSn system
- 32-atom CrFeCoNi system
- 32-atom AlCrFeCoNi system

For the above four systems we consider the cost of exploring the composition space. For the purpose of illustration, we presume that we wish to predict the electron density for 1000 compositions, with 10 multiple distinct configurations each (disordered alloy properties for a given composition are often required to be obtained as averages over several configurations). Given that, the cost of exploration using KS-DFT will be:

$$\text{Cost}_{DFT} = N_C \times N_S \times C_{DFT} \tag{1}$$

where, N_C is the number of compositions, N_S is the number of configurations per composition and C_{DFT} is the computational cost of a single electron density calculation via KS-DFT.

The cost of exploration using ML approach will be:

$$\text{Cost}_{ML} = \text{Cost}_{\text{Data}} + \text{Cost}_{\text{Training}} + \text{Cost}_{\text{Inference}} \tag{2}$$

where, $\text{Cost}_{\text{Data}}$ is the training data generation cost, $\text{Cost}_{\text{Training}}$ is the cost of training the ML model and $\text{Cost}_{\text{Inference}}$ is the cost of prediction from the ML model. The cost for training data generation can be calculated using Equation 1 based on the number of compositions and configurations used for training only. The cost of prediction however, will involve prediction at all compositions and all configurations.

To illustrate our claim of improved efficiency in more quantitative terms, Table S7 compares the cost of the KS-DFT approach vs. the ML approach, as determined from our experiences with data generation and model training in this study. (Note that the ML model trained using 64-atom SiGeSn data is used for 216-atom SiGeSn system as well.) It is evident that the 'total cost' values shown in Table S7 are dependent on the number of system snapshots to be processed (where the number of snapshots is equal to 'number of compositions to obtain electron densities for' times 'number of configurations at each composition'). At lower snapshot quantities, the KS-DFT approach is more efficient, because the cost of training a model presents a higher upfront cost. However, the cost of using the ML model to predict the density is much cheaper than the cost of performing a KS-DFT calculation. Thus, even though the KS-DFT approach is cheaper at smaller snapshot quantities, as the number of snapshots for which to obtain electron densities increases, the efficiency of the ML approach will outperform that of KS-DFT. Also note that for the sake of a fair comparison, all computations relevant to obtaining these costs were carried out on CPUs (for both KS-DFT and ML approaches). In realistic deployment scenarios, ML training, testing and inference would be carried out GPUs, thus making the large performance gains of the ML based approach even more likely.

In line with the above discussion, for each system, we observe a *crossover point* where the ML approach becomes less computationally costly than the KS-DFT approach. Figure S12 displays these crossover points for the four models that we trained. The location of the crossover point depends primarily on how expensive one KS-DFT calculation is for that system, and also on how expensive the ML model is to train. Since the 64-atom SiGeSn, owing to soft pseudopotentials and coarser KS-DFT calculation meshes, has a very low computational cost for each KS-DFT calculation (as listed in Table S7), it takes a larger number of snapshots before the ML approach overtakes the KS-DFT approach in terms of computational efficiency. Additionally, Figure S13 consolidates the subplots of Figure S12 into a single plot, to enable a direct comparison of efficiency gains from scalability. We emphasize that the values provided in Table S7, Figure S12, and Figure S13 reflect the specific models that we trained, and these values could change depending on pseudopotential choices, KS-DFT calculation parameters, model training choices, and so forth. Nevertheless, we believe that these examples are illustrative of the efficiency gains from the ML approach.

Based on the above discussion, and upon looking over the crossover points shown in Figure S12, we are led to the fact that the relative efficiency of the KS-DFT and ML approaches for compositional exploration is ultimately dependent on how many compositions are required for exploration of the composition space for a given alloy system. Table S8 addresses this question. It is clear that the number of alloy compositions scales rapidly with the number of species, as well as fineness with which the composition space is sampled (i.e., the percentage increments in each elemental concentration), leading to a combinatorial explosion that renders exhaustive first-principles exploration computationally prohibitive. We also remark that since our KS-DFT calculations do not use any kind of statistical averaging over atomic configurations, the simulation supercell size directly constrains the achievable percentage increments. Smaller increments require larger supercells, which in turn increase computational cost due to the cubic scaling of KS-DFT with system size. As a result, the ML-based exploration approaches presented here become even more attractive as the number of species grows and/or the desired resolution in compositional space increases.

	Comparison Category	32-atom CrFeCoNi Example	64-atom SiGeSn Example	216-atom SiGeSn Example	32-atom AlCrFeCoNi Example
Number of compositions to obtain electron densities for	KS-DFT	1000	1000	1000	1000
Number of configurations at each composition	KS-DFT	10	10	10	10
Cost of a single electron density calculation, via KS-DFT (in CPU hours)	KS-DFT	0.025	0.005	0.12	0.03
Total cost of KS-DFT approach (in CPU hours)	**KS-DFT**	**250**	**50**	**1200**	**300**
Number of compositions to obtain training data for	ML	10	6	6	16
Number of configurations at each training composition	ML	112	24	24	11
Cost of a single electron density calculation, via KS-DFT (in CPU hours)	ML	0.025	0.005	0.12	0.03
Total cost of training data generation (in CPU hours)	ML	*28*	*0.72*	*0.72*	*5.28*
Number of compositions to predict electron densities for	ML	1000	1000	1000	1000
Number of configurations at each prediction composition	ML	10	10	10	10
Cost of a single electron density prediction (in CPU hours)	ML	0.0025	0.0028	0.0039	0.0014
Total cost of prediction (in CPU hours)	ML	*25*	*28*	*39*	*14*
Cost of model training (in CPU hours)	ML	25	29	29	14
Total cost of ML approach (in CPU hours)	**ML**	**78**	**57.72**	**68.72**	**33.28**

Supplementary Table S7. Comparison of the computational cost between exploring the composition space of different alloy systems via the KS-DFT and ML approaches. For the sake of a fair comparison, all computations relevant to obtaining these costs were carried out on CPUs (for both KS-DFT and ML approaches).

	3-element *(e.g. SiGeSn)*	**4-element** *(e.g. CrFeCoNi)*	**5-element** *(e.g. AlCrFeCoNi)*
Number of compositions for increment of 20%	21	56	126
Number of compositions for increment of 10%	66	286	1,001
Number of compositions for increment of 5%	231	1,771	10,626
Number of compositions for increment of 1%	5,151	17,6851	4,598,126
Number of compositions for increment of 0.1%	501,501	167,668,501	42,084,793,751

Supplementary Table S8. Number of unique alloy compositions needed to fully map the composition space for different values of increment of the alloying element concentration. This count includes the sub-systems where one or more elements have 0% concentration. Multiply these numbers by number of unique configurational snapshots in order to obtain the number of snapshots shown in the x-axis of Figure S12.

G. SAD Baseline Details

In order to appreciate the accuracy of the model predictions obtained in this study, it is helpful to compare results with suitable baselines. Indeed, it is quite common in the ML literature to compare new models against community-accepted baselines, so as to standardize demonstrated performance improvements. However, at present, there is no community-accepted baseline for ML models that predict the electron density. Additionally, we are also not aware of other ML models that predict electron density across composition space for the ternary and quaternary alloys considered here; this work is novel in that regard. Thus, given that our work is proposing an entire framework, rather than simply improving upon an existing approach, obvious baselines are not readily available. To address this issue, we have selected the superposition of atomic densities (SAD) [8, 9] to serve as a baseline model for electron density prediction. Additionally, since our ML models across composition space are developed using Bayesian Active Learning, they are compared against models developed by random selection of compositions. Each of these are discussed further below.

Our reason for choosing SAD as the baseline model for electron density prediction is that it has long been recognized [10–12] to capture a good fraction of the actual electron density in various systems (a recent study [9] estimates it to be ∼85% accurate in getting the electron density of molecular systems). Furthermore, SAD are inexpensive to compute (no KS-DFT calculations are needed). Since our ML models are trained on KS-DFT data, which *do* contain atomic bonding information, while the SAD *do not*, the SAD make for a convenient baseline. Notably, as mentioned in the main text, we already utilize SAD in our work; we provided an example of a CrFeCoNi model trained on the difference between electron density and SAD (i.e. $\rho-$SAD) and found that this approach was effective for error reduction.

To obtain our own assessment of the "baseline" error that results from just using the SAD, we compute this quantity as a field over the grid points (which we already had to do when using our $\rho-$SAD model training approach). Thereafter, we treat it as if the values were predicted by an ML model, and calculate the density errors and post-process the field to obtain energy errors. We performed this analysis for 455 snapshots across the 69 compositions that comprised the quaternary CrFeCoNi dataset. The results are displayed in Fig. 9 of the main text. The performance improvements of our ML models compared to the SAD baseline are evident.

To get a sense of the baseline errors while predicting across composition space, and to demonstrate the advantage of the Bayesian AL technique over the random selection of compositions, we have compared the errors from these two approaches in Fig. 14 of the main text. In this comparison, both approaches used the same number of compositions and the same amount of data. The advantage of the Bayesian AL technique is evident from the error plot. Three different sets of 20 randomly chosen compositions were used to develop three ML models and their errors are shown in the Fig. 14 of the main text (error bars indicate the range of maximum NRMSE values observed across the three models).

[1] S. Pathrudkar, P. Thiagarajan, S. Agarwal, A. S. Banerjee, and S. Ghosh, npj Computational Materials **10**, 175 (2024).

[2] K. Jacob, S. Raj, and L. Rannesh, International Journal of Materials Research **98**, 776 (2007).

[3] C. Barrett and T. Massalski, *Structure of Metals* (McGraw-Hill, 1966) pp. Appendix VII, pg. 552.

[4] P. Hirel, Computer Physics Communications **197**, 212 (2015).

[5] C. Chen, Y. Zuo, W. Ye, X. Li, and S. P. Ong, Nature Computational Science **1**, 46 (2021).

[6] C. Chen and S. P. Ong, Nature Computational Science **2**, 718 (2022).

[7] Y. S. Teh, S. Ghosh, and K. Bhattacharya, Mechanics of Materials **163**, 104070 (2021).

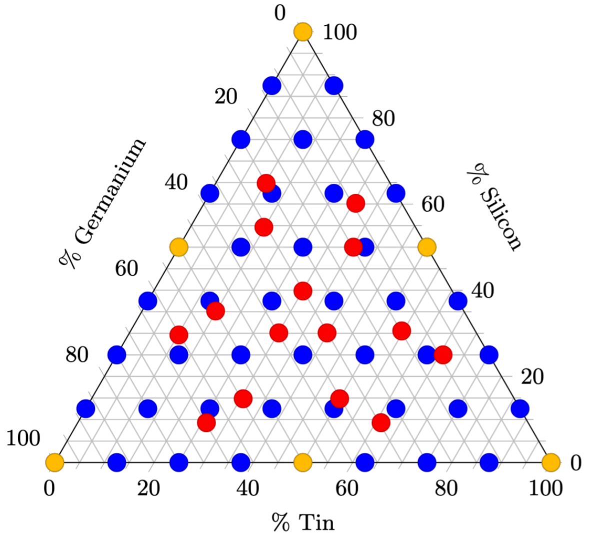
[8] P. B. Jørgensen and A. Bhowmik, npj Computational Materials **8**, 183 (2022).

[9] C. Li, O. Sharir, S. Yuan, and G. K.-L. Chan, Nature Communications **16**, 4811 (2025).

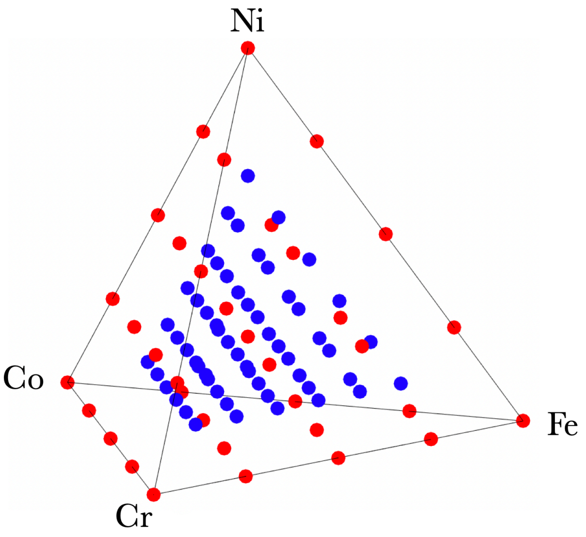
[10] A. Read and R. Needs, Journal of Physics: Condensed Matter **1**, 7565 (1989).

[11] W. Foulkes, Physical Review B **48**, 14216 (1993).

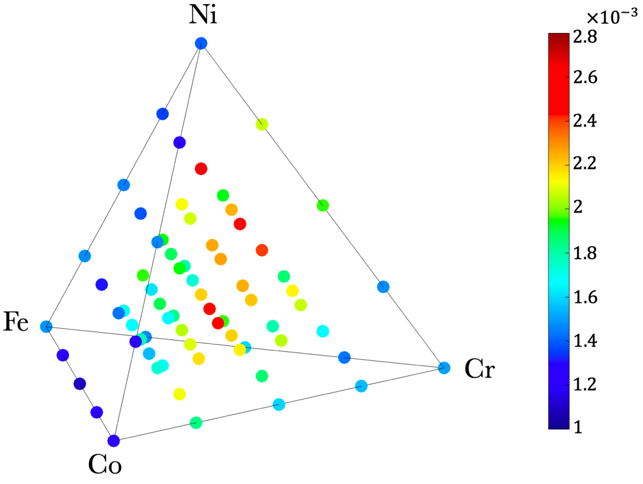
[12] G. D. Bellchambers and F. Manby, The Journal of chemical physics **135** (2011).



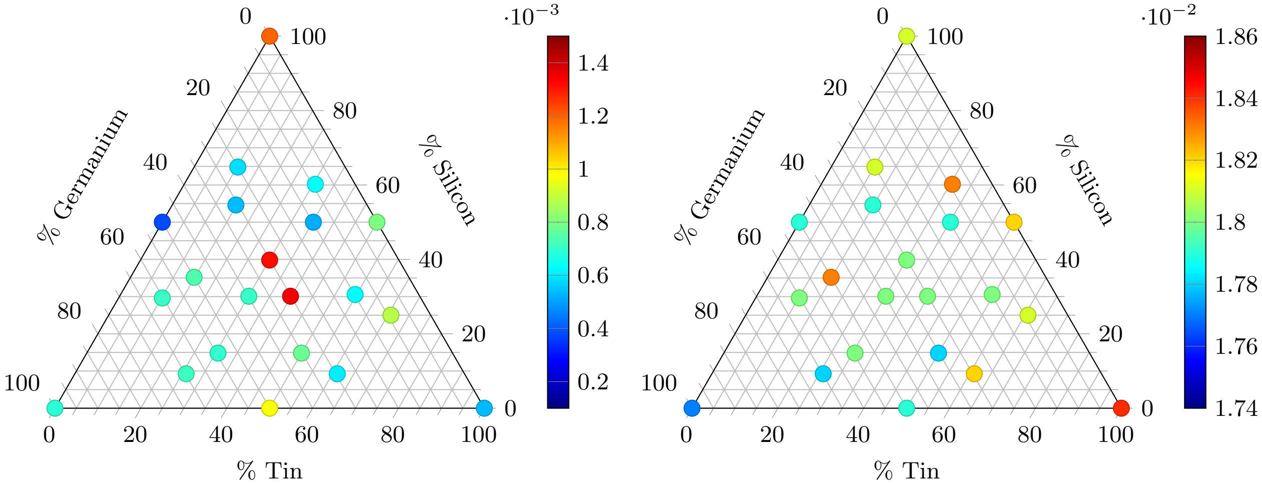
Supplementary Figure S1. **Training and Testing compositions in the SiGeSn data set.** Points in blue indicate the 64-atom data set. Note that, only a subset of compositions are used for training. Points in red indicate the 216-atom data set (not used in training). Points in yellow indicate compositions present in both the 64-atom and 216-atom data sets. See **Tables S1** and **S2** for the full list of values.



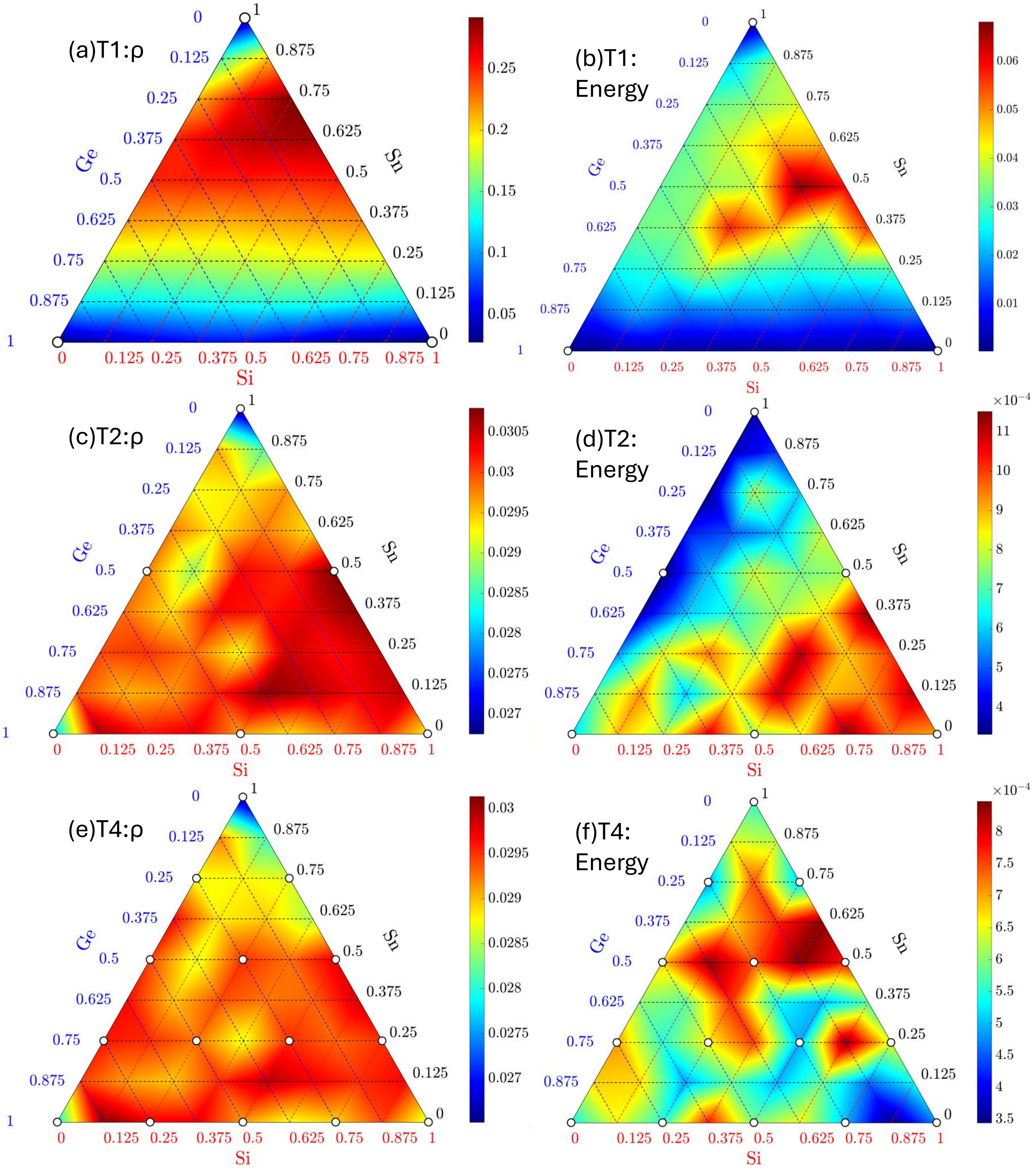
Supplementary Figure S2. **Training and testing compositions in the 32-atom CrFeCoNi data set**. Points in blue indicate the true quaternary compositions, while points in red indicate ternary, binary, and unary derivatives. Note that, only a subset of compositions are used for training. See **Table S3** for the full list of values.



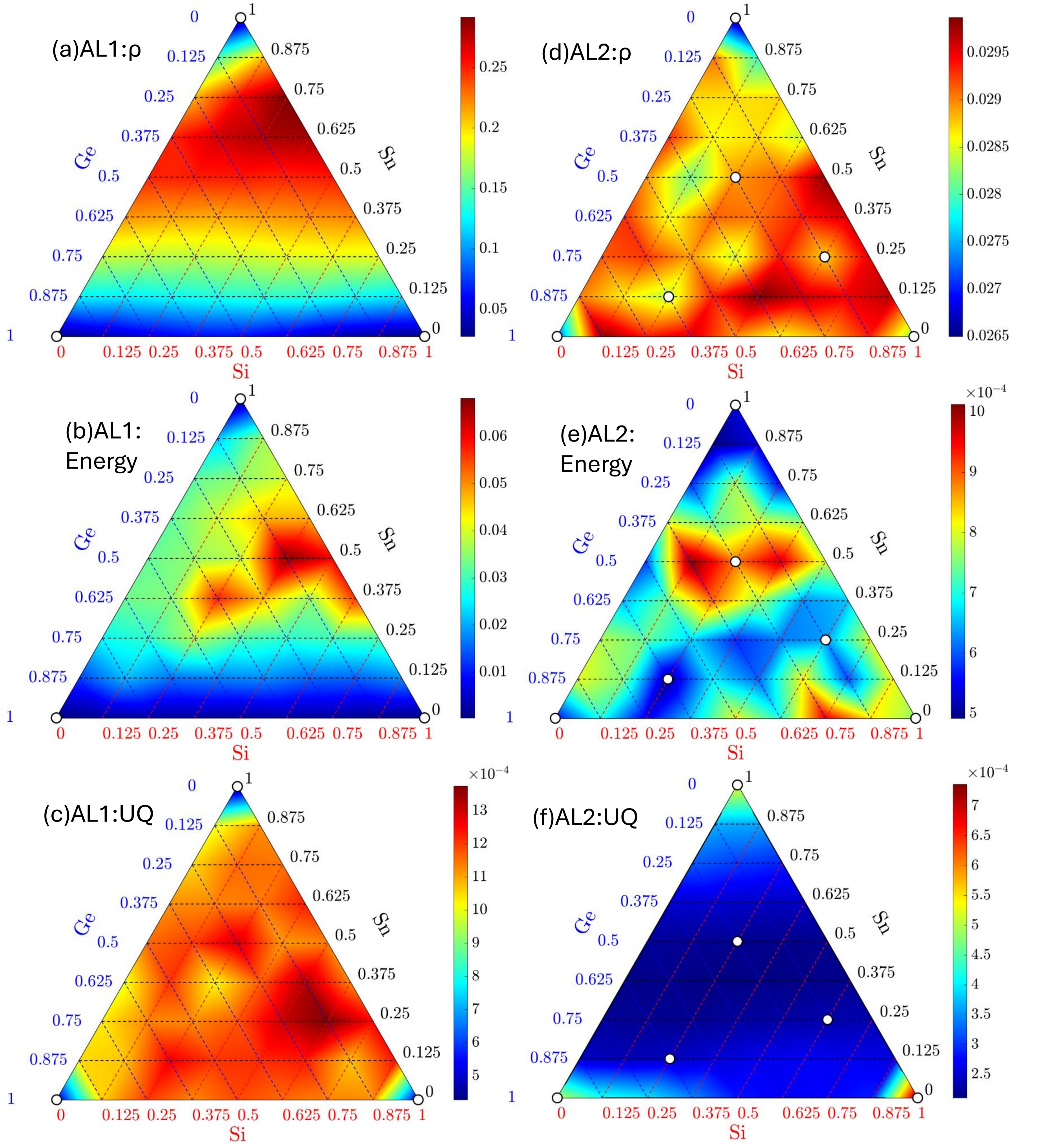
Supplementary Figure S3. **Error in energy for the quaternary CrFeCoNi system for the model trained with charge density ρ instead of $\delta\rho$.** The plot shows the average error in energy at test compositions for the pristine 32-atom CrFeCoNi data set, expressed in Hartree per atom (Ha/atom). The order of magnitude of the colorbar is 10^{-3}.



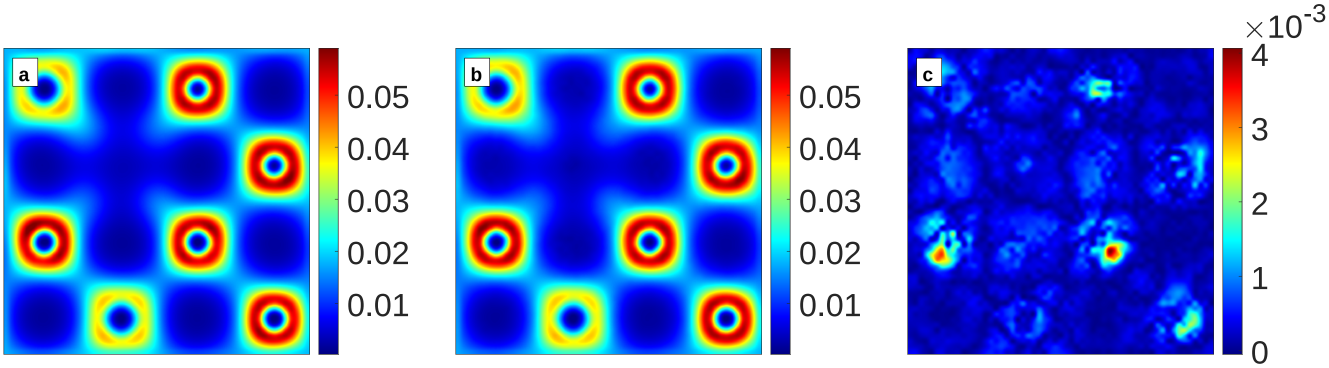
Supplementary Figure S4. **Error for ternary system beyond training compositions and training system size.** With a 64 atom ternary system, a limited number of test compositions is possible. We used a bigger 216-atom system to obtain intermediate test compositions, not possible with the 64 atom system. *Left:* Average errors in energy at test compositions for the pristine 216-atom SiGeSn data set, using the AL2 model. Units: Hartree/atom. Note that the order of magnitude of the colorbar is 10^{-3}. *Right:* Average error in density at test compositions for the pristine 216-atom SiGeSn data set, in terms of relative L1, using the AL2 model. Note that the order of magnitude of the colorbar is 10^{-2}.



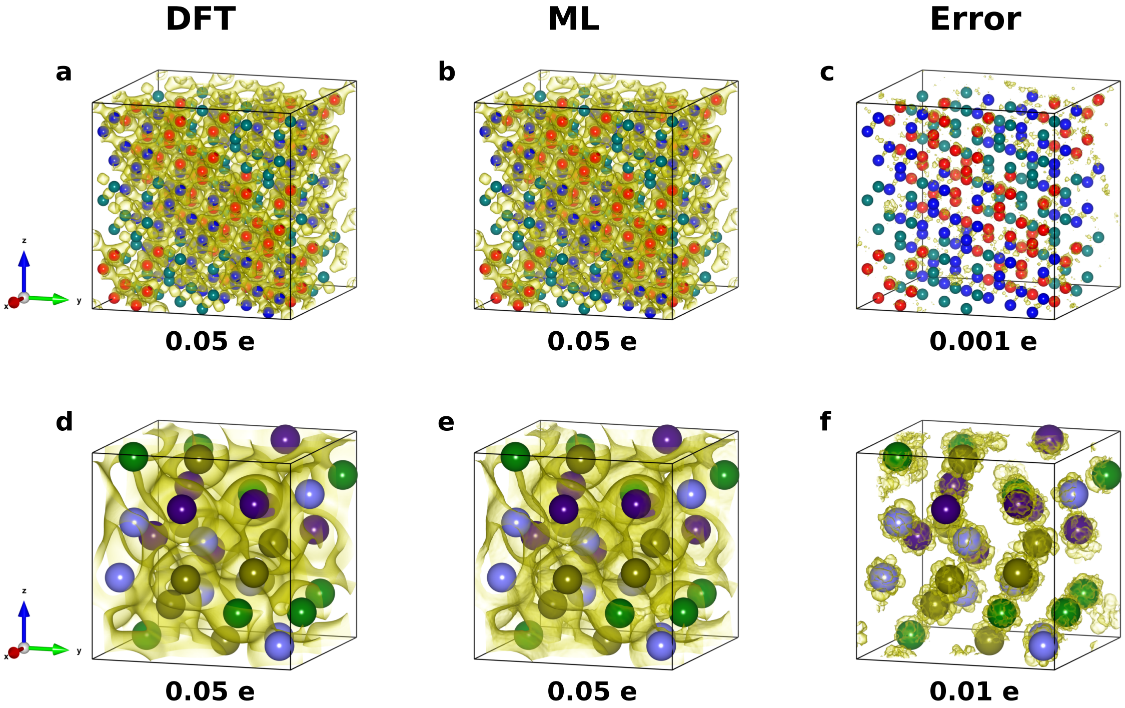
Supplementary Figure S5. **Tessellation based approach to iteratively select training compositions to accurately predict across composition space of Ternary alloy.** (a) NRMSE across the composition space after 1st iteration of Tessellation based iterative learning. The model is trained using only 3 pure compositions shown using white circles in the figure. This model is termed as T1. (b) Energy prediction error across the composition space after 1st iteration of Tessellation. This model, termed as T1, is trained only for 3 pure compositions. (c) NRMSE across the composition space after 2nd iteration of Tessellation. Three additional training points. This model is termed as T2. (d) Error in energy prediction across composition space. The unit of energy error is Ha/atom. The predicted energy is obtained from ρ predictions from T2. The energy error is within chemical accuracy across the composition space. (e) NRMSE across the composition space after 4th iteration of Tessellation, resulting in nine additional training points from T2. This model is termed as T2. (f) Energy prediction error across the composition space after 4th iteration of Tessellation. This model, termed as T4, is trained only for 15 compositions.



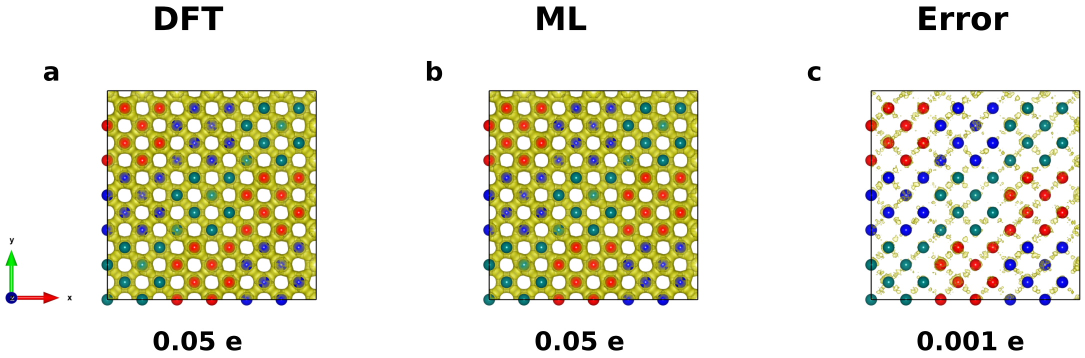
Supplementary Figure S6. **Bayesian Active Learning to iteratively select training compositions to accurately predict across composition space of Ternary alloy.** (a) NRMSE across the composition space after 1st iteration of Active Learning, termed as AL1, trained using only 3 pure compositions shown using white circles. (b) Energy prediction error for model AL1 with 3 pure composition. (c) Epistemic Uncertainty in ρ prediction across composition space after prediction with model AL1. Query points (additional training points) for the next iteration of Bayesian Active Learning are selected based on highest uncertainty regions shown in ‘f’. (d) NRMSE across the composition space after 2nd iteration of Active Learning. 3 additional training points are added as per the uncertainty contour in subfigure, ‘c’. This model is termed as AL2. We observe that the NRMSE is low and consistent across the composition space showing the effectiveness of query points selection through uncertainty. (e) Error in energy prediction across composition space. The unit of energy error is Ha/atom. The predicted energy is obtained from ρ predictions from AL2. The energy error is within chemical accuracy across the composition space. (f) Epistemic Uncertainty in ρ prediction across composition space after prediction with model AL2.



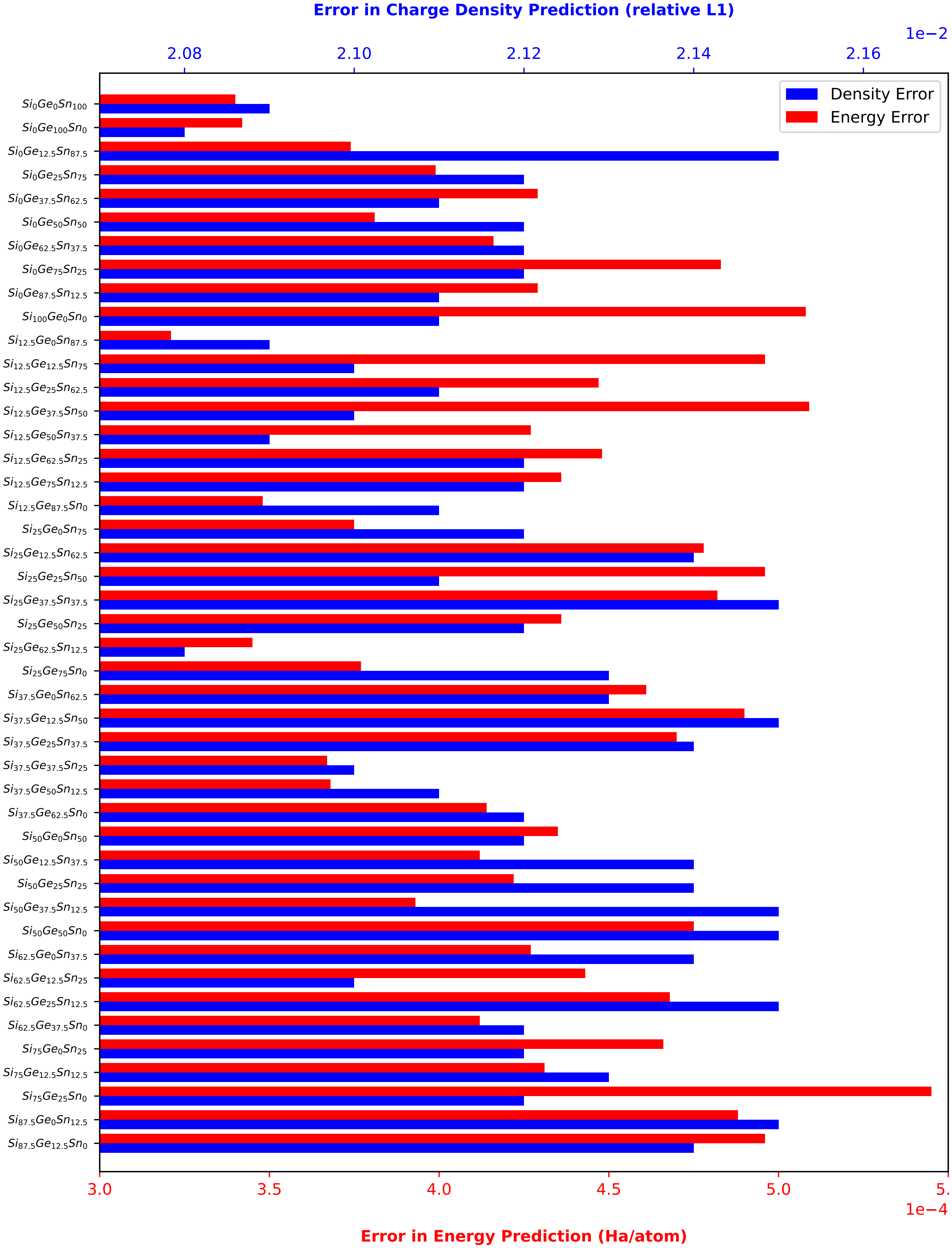
Supplementary Figure S7. Electron densities (a) calculated from KS-DFT and (b) predicted by ML and the absolute difference between them (c) for a vacancy defect for SiGeSn. The snapshot corresponds to 64 atom $Si_{29.7}Ge_{29.7}Sn_{40.6}$ simulation cell at 2400K with an Sn vacancy. The ternary AL2 model was used here.



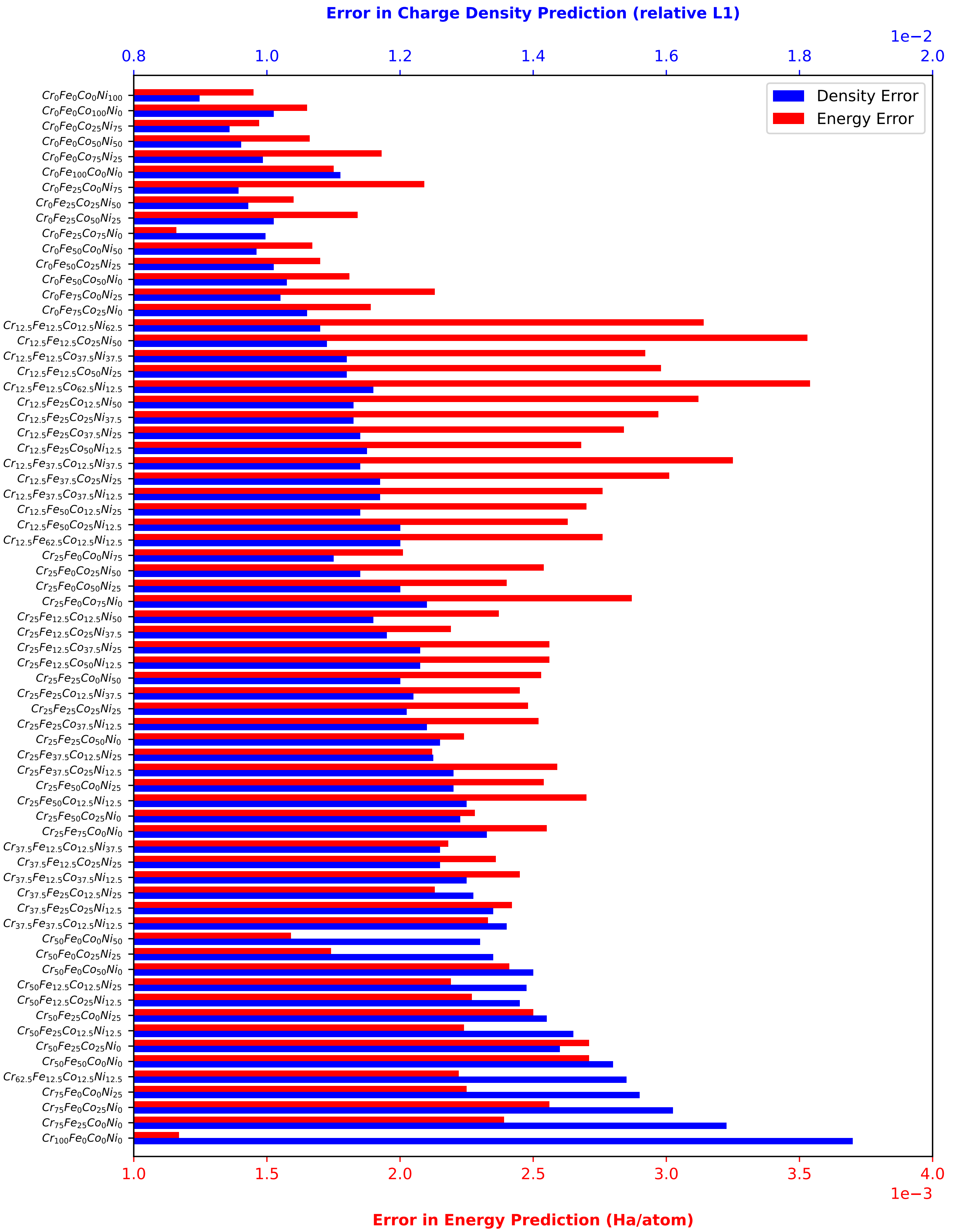
Supplementary Figure S8. Electron densities (a, d) calculated by KS-DFT and (b, e) predicted by ML and the Error (absolute difference) between them (c, f) for SiGeSn (a, b, c) and CrFeCoNi (d, e, f), using the AL2 model. Subplots (a, b, c) correspond to a 216-atom $Si_{33.3}Ge_{33.3}Sn_{33.3}$ simulation cell at 2400K for the handcrafted systems featuring species segregation. Subplots (d, e, f) are 32-atom simulation cells at 5000K corresponding to $Cr_{25}Fe_{25}Ni_{25}Co_{25}$ for the ρ model. The values refer to the iso-surface values.



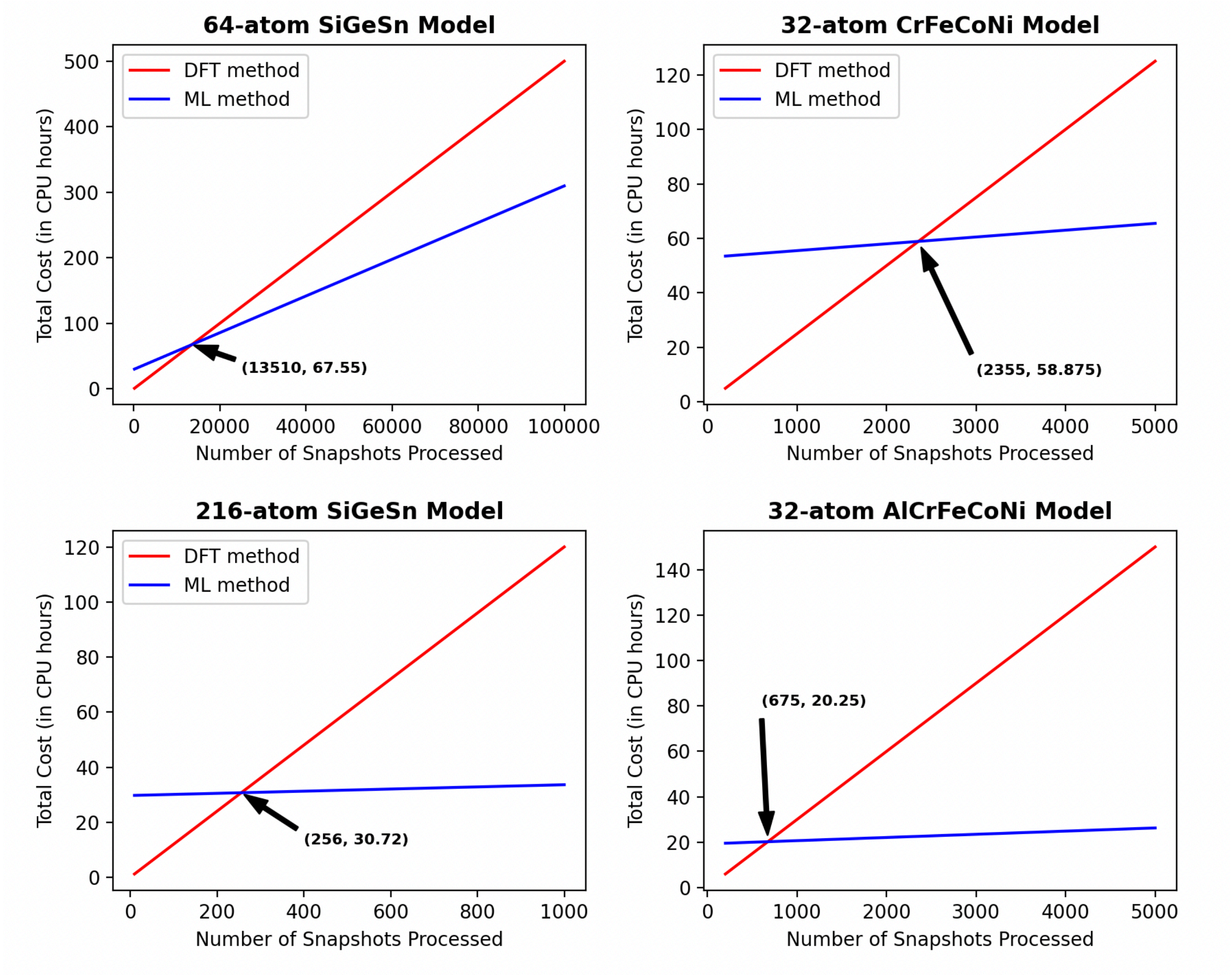
Supplementary Figure S9. Electron densities (a) calculated by KS-DFT and (b) predicted by ML and the Error (absolute difference) between them (c) for SiGeSn (a, b, c) using the AL2 model. Subplots (a, b, c) correspond to a 216-atom $Si_{33.3}Ge_{33.3}Sn_{33.3}$ simulation cell at 2400K same as Figure S8 for the handcrafted systems featuring species segregation visualized in the xy plane. The values refer to the iso-surface values. Blue, red and turquoise spheres represents Si, Ge, and Sn atoms, respectively.



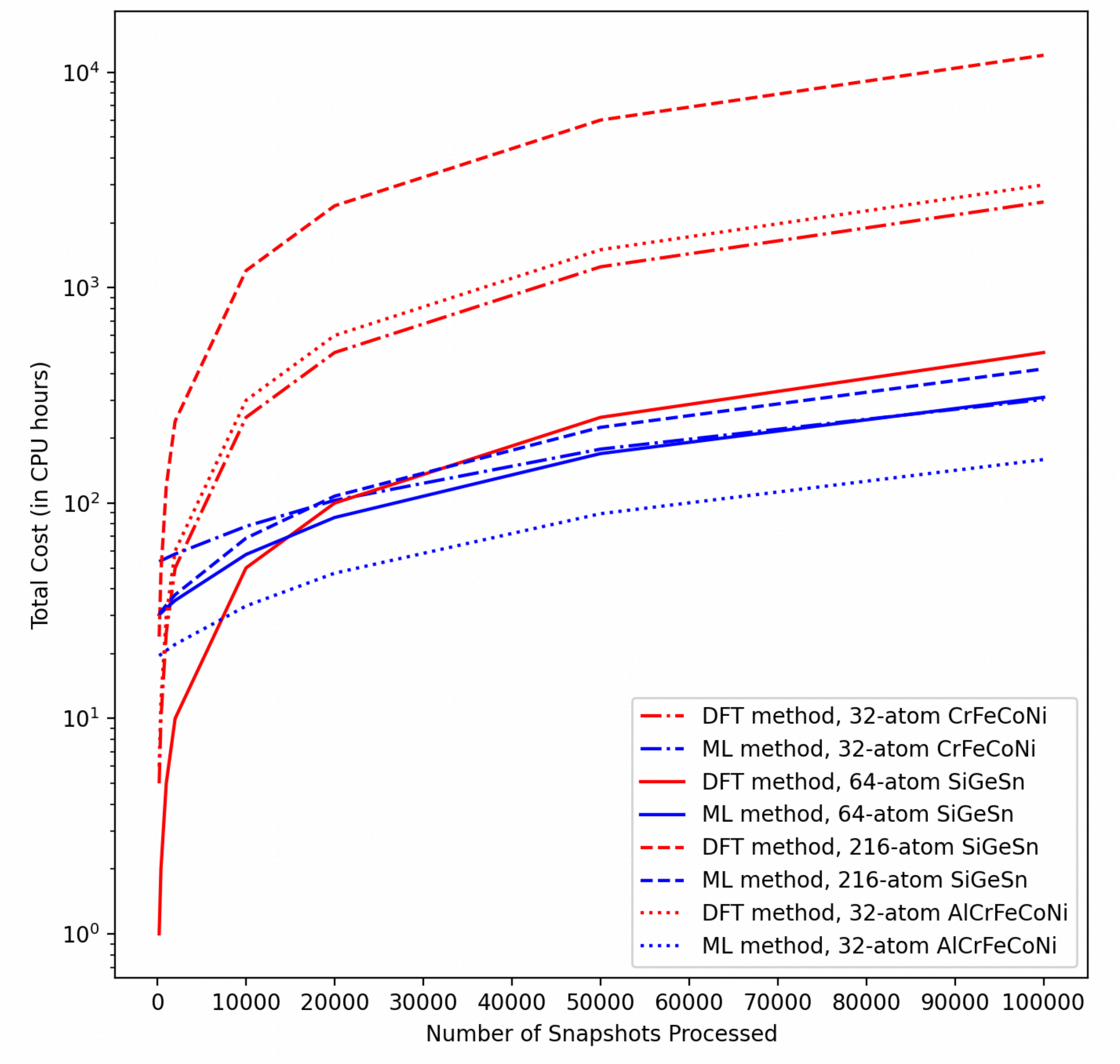
Supplementary Figure S10. Plot of average energy and density errors, by composition, for the pristine 64-atom SiGeSn test data. The ternary AL2 model was used here. The results are averaged over all snapshots available for each given composition. 🟦: Density Error, 🟥: Energy Error



Supplementary Figure S11. Plot of average energy and density errors, by composition, for the pristine 32-atom CrFeCoNi test data (trained on the difference between the charge density field and the atomic densities). The quaternary AL3 model was used here. 🟦: Density Error, 🟥: Energy Error



Supplementary Figure S12. **Snapshots vs. Total Cost.** These subplots show the crossover point where the ML method becomes less computationally costly compared to the KS-DFT method. The primary cost of the ML method is the upfront cost of training the model. The crossover point arises since the cost of generating a prediction from the model is much cheaper that the cost of performing a full KS-DFT calculation. Each system will have a different crossover point, depending on the expense of a KS-DFT calculation for that system and on the setup choices for the ML model. The crossover points shown here are for four example systems that were considered in this work; the assumptions made to produce these subplots are shown in Table S7. The crossover point for the 64-atom SiGeSn system is higher due to the low cost of the KS-DFT calculation; the number of valence electrons considered was minimal due to the p-block location of the elements. The number of snapshots shown on the x-axis of these subplots is obtained by multiplying the number of compositions by the number of configurational snapshots at each composition. An overview of the number of compositions that would be needed to explore the composition space in different increment sizes is shown in Table S8. Additionally, for a direct comparison of all four systems, Figure S13 consolidates these four subplots into a single plot. Note that, for the sake of a fair comparison, all computations relevant to obtaining these costs were carried out on CPUs (for both KS-DFT and ML approaches). —: ML method, —: DFT method



Supplementary Figure S13. **Consolidated Plot.** This graph presents the data from Figure S12 into a single logarithmic plot to allow for more direct comparison between systems. Notice that, in each case, the ML costs (shown in blue) have a higher starting value, but they soon overtake the KS-DFT costs (shown in red) in terms of efficiency as the number of snapshots to obtain electron densities for increases. Note that, for the sake of a fair comparison, all computations relevant to obtaining these costs were carried out on CPUs (for both KS-DFT and ML approaches). -·-·-: DFT method, 32-atom CrFeCoNi, -·-·-: ML method, 32-atom CrFeCoNi, ——: DFT method, 64-atom SiGeSn, ——: ML method, 64-atom SiGeSn, - - -: DFT method, 216-atom SiGeSn, - - -: ML method, 216-atom SiGeSn, ······: DFT method, 32-atom AlCrFeCoNi, ······: ML method, 32-atom AlCrFeCoNi.