Architectural Blueprint for a High-Fidelity Machine Learning Inorganic Reaction Prediction Engine on macOS Silicon

1. Executive Summary and Strategic Alignment

The successful simulation of chemical reactivity in a computational environment represents one of the grand challenges of modern materials informatics. The objective detailed in this report is to architect and implement a local, machine-learning-driven reaction prediction mechanism tailored for a specific front-end interface—the Evolutionary Origin of Hierarchy (EOH) platform. This mechanism must accept binary elemental inputs from a pre-determined selection, predict the thermodynamically stable reaction products, and operate efficiently on a local macOS environment utilizing Apple Silicon. Furthermore, the architecture must be extensible, designed from the outset to accommodate future thermodynamic variables such as pressure (P) and temperature (T).

Unlike organic chemistry, which is often governed by kinetic controls and molecular graph isomorphism, inorganic binary reaction prediction is fundamentally a problem of thermodynamic phase stability. The reaction between two elements, A and B, is mathematically equivalent to identifying the Global Minimum on the Gibbs Free Energy surface of the $A-B$ system. Consequently, the machine learning task is not one of simple classification (reacts vs. does not react) but rather a high-precision regression task: predicting the formation enthalpy (ΔH_f) of all possible stoichiometries to construct a Convex Hull of stability.

This report provides an exhaustive analysis of the requisite components for building this engine. It begins with a deep theoretical treatment of the thermodynamic principles that the machine learning model must approximate. It then proceeds to a comprehensive evaluation of the data landscape, dissecting the nuances of the Materials Project, OQMD, and experimental datasets to empower the user to make an informed selection. The report evaluates state-of-the-art model architectures—specifically comparing Random Forests, Graph Neural Networks (Roost), and Transformer-based models (CrabNet)—and recommends a specific implementation path optimized for the Unified Memory Architecture of Apple Silicon (M1/M2/M3 chips). Finally, it details the precise logic for integrating these predictions into the EOH front end and outlines a rigorous roadmap for incorporating temperature and pressure variables via CALPHAD (Calculation of Phase Diagrams) integration.

---

2. Theoretical Foundation: The Physics of Binary Reactivity

To accurately simulate the "combination of two elements" as requested, the underlying machine learning model must mimic the physical laws governing solid-state reactions. When a user on the EOH front end drags "Element A" to "Element B," they are effectively initiating a virtual annealing process. The computational engine must determine the equilibrium state of this mixture. This section delineates the physical theory that frames the machine learning problem.

2.1 Thermodynamic Stability and the Convex Hull Construction

In the context of inorganic synthesis, a "reaction" is defined as the transformation of reactants into products that lowers the total free energy of the system. For solid-state systems at standard conditions (typically approximated at 0 K in Density Functional Theory datasets), the governing potential is Enthalpy (H). The fundamental metric for prediction is the **Formation Energy** (ΔE_f), which represents the energy change associated with forming a compound from its constituent elements in their standard reference states.

The mathematical formulation for the formation energy of a binary compound $A_x B_y$ is given by:

$$\Delta E_f(A_x B_y) = E_{total}(A_x B_y) - x E_{bulk}(A) - y E_{bulk}(B)$$

where E_{total} is the DFT-calculated total energy of the compound, and E_{bulk} represents the energy per atom of the elemental reference states.1 A negative formation energy indicates that the compound is energetically favored over the isolated elements. However, a negative formation energy is a necessary but insufficient condition for a reaction to occur. A compound must also be stable against decomposition into other compounds within the same chemical system.
This stability is determined geometrically by the **Convex Hull**. When the formation energies of all possible compounds in the $A-B$ binary system are plotted against the composition fraction of B ($x_B = y / (x+y)$), the convex hull is defined as the lower envelope of these points.

- **Stable Phases:** Any compound whose energy lies exactly *on* the convex hull is thermodynamically stable. These are the predicted reaction products.
- **Metastable/Unstable Phases:** Any compound whose energy lies *above* the hull is unstable. It will spontaneously decompose into the stable phases that define the hull at that composition. The vertical distance from a point to the hull is termed the "Energy

Above Hull" (E_{hull}), a critical metric for quantifying instability.[4]

Implication for Machine Learning: The "Reaction Prediction Mechanism" is, therefore, a **Phase Diagram Generator**. The ML model does not directly predict "Reaction: Yes/No." Instead, it acts as a surrogate for Density Functional Theory (DFT). It must predict the ΔE_f for a sweep of theoretical stoichiometries (e.g., AB, AB_2, A_2B, A_3B_4). The engine then feeds these predictions into a convex hull construction algorithm (available in libraries like pymatgen) to determine which of these theoretical compounds define the ground state.[5]

2.2 The Challenge of Structure-Agnostic Prediction

A critical constraint of the EOH front end is that the input consists solely of elemental identities (e.g., "Iron" and "Oxygen") and potentially a mixing ratio. The user does not provide the crystal structure (lattice vectors, atomic coordinates) of the potential product.

This constraint fundamentally dictates the class of machine learning models that can be employed. The vast majority of modern materials informatics models, such as Crystal Graph Convolutional Neural Networks (CGCNN) or Materials Graph Network (MEGNet), are **Structure-Based**. They require the detailed crystal structure as input to predict energy.[1] These models are useless for *ab initio* reaction prediction because the product structure is the *unknown* variable we wish to discover.

Consequently, the proposed mechanism must utilize **Composition-Based Feature Vectors (CBFVs)** or **Stoichiometry-to-Structure** architectures. Composition-based models learn a mapping function $f: \text{Composition} \rightarrow \text{Energy}$ by aggregating the chemical properties of the constituent elements (electronegativity, atomic radius, valence electrons) without explicit knowledge of the atomic positions. Recent advancements, such as the CrabNet (Compositionally Restricted Attention-Based Network) and Roost (Representation Learning from Stoichiometry), have demonstrated that deep learning architectures using self-attention can achieve prediction accuracies rivaling structure-based methods, making them the ideal candidates for this application.[8]

---

3. Data Strategy: Exhaustive Dataset Selection and Analysis

The predictive capability of the machine learning engine is strictly bounded by the quality, diversity, and coverage of the training data. For inorganic chemistry, high-throughput Density Functional Theory (HT-DFT) databases are the gold standard. The user has requested a presentation of all viable options to make a final decision. The following section provides a detailed comparative analysis of the primary data ecosystems suitable for binary reaction

prediction.

3.1 Option A: The Materials Project (MP) - The Gold Standard

The Materials Project (MP) is the most widely utilized database in the materials informatics community. It contains computed properties for over 150,000 inorganic compounds, calculated using the Vienna Ab initio Simulation Package (VASP).[11]

- **Data Provenance:** MP uses the Generalized Gradient Approximation (GGA) with the PBE functional. Crucially, it employs "Hubbard U" corrections (DFT+U) for transition metal oxides to correct for self-interaction errors, ensuring accurate energy predictions for materials like rust (Iron Oxide) or battery cathodes.[13]
- **Binary Coverage:** MP has excellent coverage of binary systems, containing calculations for the vast majority of experimentally known binaries as well as thousands of theoretical structures.
- **Infrastructure Synergy:** The strongest argument for using MP is its tight integration with pymatgen (Python Materials Genomics). The mp-api (formerly MAPI) allows for seamless querying of "entries" that are pre-formatted for phase diagram construction. The compatibility schemes—algorithms that adjust energies to mix different types of calculations—are native to this ecosystem.[4]
- **Suitability for EOH:** High. The existence of the get_entries_in_chemsys endpoint means the data extraction pipeline is robust and requires minimal cleaning. The data is consistent, minimizing the noise the ML model interprets.

3.2 Option B: The Open Quantum Materials Database (OQMD) - The Theoretical Vastness

The OQMD, developed by the Wolverton group at Northwestern University, takes a different philosophical approach. While MP focuses on "known or likely" materials, OQMD generated data by systematically decorating crystal prototypes with every possible combination of elements. This results in a database of over 1,000,000 entries.[15]

- **The "Negative Data" Advantage:** A significant challenge in training reaction predictors is "survivorship bias." If a model is trained only on stable compounds (MP), it may learn that *everything* is stable. OQMD contains hundreds of thousands of **unstable** compounds (high energy above hull). Including this data is critical for teaching the ML model what *not* to predict. It provides the "negative examples" necessary for a robust regression boundary.[3]
- **Thermodynamic Focus:** OQMD was built specifically for thermodynamics and convex hull construction. Its reference states are rigorously defined for this purpose.
- **Cons:** There are systematic offsets between OQMD and MP energies due to different pseudopotentials and U-values.[3] Mixing data from MP and OQMD without careful calibration can introduce errors.
- **Suitability for EOH:** High, specifically as a supplement to MP for "unstable" training

samples.

3.3 Option C: AFLOW (Automatic-FLOW) - The Alloy Specialist

AFLOW is a massive repository (3.5 million+ entries) with a strong emphasis on metallic alloys, intermetallics, and Heusler compounds.[11]

- **Strengths:** Unrivaled coverage of metal-metal binary systems. If the EOH front end is expected to handle a lot of metallurgy or alloy mixing (e.g., mixing Copper and Zinc), AFLOW provides the densest training data.
- **Weaknesses:** The API is generally considered less user-friendly ("Pythonic") than MP or OQMD. The sheer volume of data can be overwhelming and contains many redundant calculations that require aggressive deduplication.
- **Suitability for EOH:** Moderate. Best reserved if the user's specific domain interest lies in metallurgy.

3.4 Option D: JARVIS-DFT - The Accuracy Specialist

Maintained by NIST, JARVIS uses the optB88vDW functional, which accounts for Van der Waals forces. This makes it significantly more accurate for 2D materials (like Graphene or MoS2) and layered structures than MP or OQMD.[17]

- **Relevance:** If the user intends to simulate exfoliation or 2D material synthesis, JARVIS is essential.
- **Suitability for EOH:** Low for a general-purpose binary predictor, as the dataset is smaller (~40,000-70,000 entries) and the different functional means energies cannot be compared to MP/OQMD. However, for the future request of **Temperature/Pressure**, JARVIS contains elastic constants and phonon data vital for entropy calculations.

3.5 Option E: Experimental Data (ICSD/NIST)

Using purely experimental databases like the Inorganic Crystal Structure Database (ICSD) or NIST standard reference data.[19]

- **Pros:** Ground truth. DFT has systematic errors (e.g., underestimating band gaps). Experimental data represents reality.
- **Cons:** Sparse. There are only ~50,000 well-characterized inorganic structures compared to millions of DFT calculations. Training deep learning models on such small data often leads to overfitting. Furthermore, experimental databases rarely contain "failed" reactions (unstable compounds), making it hard to train the stability boundary.

3.6 Data Selection Decision Matrix

To facilitate the user's final decision, the following comparison matrix summarizes the options based on the project constraints:

Feature	Materials Project (MP)	OQMD	AFLOW	JARVIS-DFT	Experimental (ICSD)
Primary Focus	General Inorganic	Thermodynamics	Alloys/Metals	2D/Electronic	Real Synthesis
Size (Entries)	~150,000	~1,000,000	~3,500,000	~70,000	~50,000
API Quality	Excellent (mp-api)	Good (qmpy)	Moderate	Good (jarvis-tools)	Low (Proprietary)
Negative Data	Low	**High**	High	Moderate	Very Low
Binary Coverage	High	Very High	Very High	Moderate	Moderate
Mac Compatibility	Native (pymatgen)	Native	Web API	Native	N/A

Recommendation: For the EOH reaction prediction mechanism, the optimal strategy is a **Hybrid Dataset**.

1. **Core Training Set:** Use **The Materials Project** for the high-quality, curated ground state structures. Its integration with the Python ecosystem on macOS is superior.
2. **Augmentation:** Supplement with **OQMD** specifically to ingest unstable (positive formation energy) binary compounds. This will teach the ML model to reject unrealistic stoichiometries.
3. **Future-Proofing:** Download **JARVIS-DFT** data now to store elastic tensor and phonon data, which will be required for the Temperature/Pressure module later.

---

4. Machine Learning Architecture: Evaluation and Selection

The "brain" of the reaction mechanism is the regression model that predicts formation energy

from composition. We evaluate three distinct architectural classes, ranging from classical ensemble methods to state-of-the-art geometric deep learning.

4.1 Architecture I: The Magpie-Random Forest (Baseline)

Before the advent of deep learning, the standard approach in materials informatics was "Feature Engineering" using the Magpie (Materials-Agnostic Platform for Informatics and Exploration) descriptors.[21]

- **Mechanism:** This approach converts a chemical formula (e.g., Fe_2O_3) into a fixed-length vector of statistical properties.
  - *Features:* The vector includes the mean, range, and variance of atomic properties of the constituent elements: atomic number, atomic mass, melting point, electronegativity, valence d-electrons, etc.
  - *Input Vector:* A 145-dimensional vector representing the stoichiometry.
  - *Model:* A Random Forest Regressor or Gradient Boosted Tree (XGBoost) is trained to map this vector to formation energy.
- **Pros:**
  - **Speed:** Training and inference are exceptionally fast, even on a CPU.
  - **Interpretability:** Feature importance analysis (SHAP values) can reveal *why* a prediction was made (e.g., "High electronegativity difference drove stability").[1]
  - **Data Efficiency:** Performs surprisingly well on small datasets (<10,000 points).
- **Cons:**
  - **Limited Capacity:** It cannot learn complex, non-linear representations of atomic interactions as effectively as neural networks. It assumes the provided physical features (electronegativity) are the *only* relevant factors.

4.2 Architecture II: Graph Neural Networks (Roost)

Graph Neural Networks (GNNs) are the standard for structure-based tasks, but for composition-only tasks, the **Roost** (Representation Learning from Stoichiometry) architecture adapts the graph concept.[8]

- **Mechanism:** Roost represents a composition as a dense, fully connected graph.
  - *Nodes:* The elements (e.g., Fe node, O node).
  - *Edges:* Every element is connected to every other element.
  - *Weights:* The stoichiometric fractions (0.4, 0.6) weight the edges.
- **Process:** The model passes "messages" along the edges, updating the representation of each element based on its neighbors. A final "pooling" operation aggregates the node states into a single material embedding.
- **Performance:** Roost significantly outperforms Magpie-Random Forest on large datasets (OQMD/MP), reducing Mean Absolute Error (MAE) from ~0.08 eV/atom to ~0.04 eV/atom.

4.3 Architecture III: Transformer-Based Attention (CrabNet)

The **Compositionally Restricted Attention-Based Network (CrabNet)** represents the current state-of-the-art for this specific problem.[9] It adapts the "Self-Attention" mechanism—famous for powering Large Language Models like GPT—to chemistry.

- **Mechanism:**
  - **Tokenization:** Elements are treated as tokens.
  - **Embeddings:** The model initializes with "Matschol" embeddings (dense vectors representing chemical knowledge).
  - **Fractional Encoding:** A unique innovation of CrabNet is how it handles stoichiometry. The fractions are processed through a separate encoder and injected into the attention mechanism. This allows the model to "attend" differently to Iron in FeO versus Iron in Fe_2O_3.
  - **Self-Attention:** The transformer layers allow the elements to contextually influence each other. The model learns that Oxygen "activates" different electron behaviors in Iron than Sulfur does.
- **Performance:** CrabNet consistently tops the leaderboards (e.g., Matbench) for formation energy prediction. It is robust, handles dilute dopants well, and scales effectively with data size.[26]

4.4 Model Selection for EOH on macOS

The recommended architecture for the EOH Reaction Prediction Mechanism is **CrabNet**.

Reasoning:

1. **Accuracy:** It offers the highest fidelity for formation energy, which is critical because small errors can incorrectly change the convex hull, predicting the wrong reaction product.
2. **Architecture Fit:** The "Attention" mechanism maps intuitively to the concept of chemical interaction.
3. **Hardware Optimization:** Transformers are matrix-multiplication heavy. This aligns perfectly with the **Apple Neural Engine (ANE)** and the GPU architecture on M-series chips, which are optimized for dense matrix operations typical of Transformers.
4. **Implementation:** CrabNet is implemented in PyTorch, which has excellent support on macOS via the MPS backend (discussed in Section 5).

---

5. Apple Silicon Implementation Strategy: The Local Stack

A key requirement of the user request is local execution on a Mac. Developing ML on Apple Silicon (M1/M2/M3) requires a departure from the traditional NVIDIA/CUDA stack. The Unified Memory Architecture (UMA) of Apple Silicon offers a distinct advantage: the CPU and GPU share the same memory pool, eliminating the costly data transfer overhead seen in

PCIe-based GPU systems.

5.1 The Software Stack: Environment Configuration

To utilize the hardware acceleration, we must rely on the **Metal Performance Shaders (MPS)** backend for PyTorch.

Step 1: Python Environment Management
Do not use the system Python. Use Miniforge, a community-driven distribution of conda optimized for ARM64 (Apple Silicon).

- *Why Miniforge?* It defaults to the conda-forge channel, which provides ARM64-compiled binaries for scientific libraries (NumPy, SciPy, pandas) that utilize Apple's **Accelerate** framework (vecLib) for BLAS operations.[28]

Step 2: The Deep Learning Framework (PyTorch vs. MLX)

- **PyTorch (MPS):** The current stable release of PyTorch includes the mps device. This maps Torch operations to the Metal API.
  - *Status:* Mature enough for CrabNet. Supports most attention layers and optimizers.
  - *Implementation:* device = torch.device("mps").[30]
- **MLX (Apple's Framework):** Apple Research recently released MLX, a NumPy-like array framework specifically for Apple Silicon.[31]
  - *Pros:* potentially faster than PyTorch on M-chips due to lower overhead.
  - *Cons:* Newer, smaller ecosystem. Porting CrabNet to MLX would require rewriting the model layers from scratch.
  - *Decision:* Stick to **PyTorch (MPS)** for Version 1.0 to leverage existing open-source implementations of CrabNet/Roost.

Step 3: XGBoost/LightGBM Nuances
If utilizing the Random Forest baseline (Architecture I), be aware that XGBoost GPU acceleration is primarily CUDA-based. On macOS, XGBoost will run on the CPU. However, the M-series CPUs are exceptionally fast at tree-based operations.

- *Installation:* conda install -c conda-forge xgboost ensures the library is linked against Apple's multi-threaded libraries.[33]

5.2 Performance Optimization on M-Series Chips

To ensure the EOH front end feels "snappy" (real-time reaction prediction), specific optimizations are required:

1. **Batch Size Tuning:** The MPS backend prefers larger batch sizes to saturate the GPU cores. When sweeping stoichiometries for the convex hull (predicting 50-100 potential compounds at once), send them as a single batch tensor rather than a loop of individual predictions.
2. **Float32 vs Float16:** The M-series GPU has native hardware support for FP16

(half-precision). Running the inference in float16 can double throughput and halve memory usage without significant loss in chemical precision.

3. **Warm-up Inference:** The first call to the MPS graph often incurs a compilation penalty. The application startup sequence should run a dummy prediction (e.g., "Li-O") to "warm up" the Metal shaders before the user interacts with the UI.

---

6. The "EOH-React" Engine: Detailed Implementation Logic

This section details the algorithmic logic that connects the user's frontend action (combining two elements) to the machine learning backend.

6.1 The Reaction Algorithm

The core problem is to determine the equilibrium state of a binary system $A-B$.

Phase 1: Input Handling

- **User Action:** User selects Element A (e.g., Lithium, Li) and Element B (e.g., Sulfur, S) on the EOH interface.
- **Backend Request:** The frontend sends a JSON payload: {"reactants":}.

Phase 2: Virtual Stoichiometry Sweep
The engine must explore the chemical space. Since we don't know what will form, we generate a grid of hypothetical compositions.

- *Generation Strategy:* Create a list of hypothetical compounds covering the entire composition range from $x_B = 0$ to $x_B = 1$.
- *Grid:* $A_{10}B_1, A_9B_1, \dots, AB, \dots, A_1B_9, A_1B_{10}$.
- *Common Ratios:* Explicitly include common integer ratios found in nature: 1:1, 1:2, 2:3, 2:1, 2:5, 3:4.
- *Result:* A list of ~50-100 composition strings (e.g., "Li10S", "Li2S", "LiS", "LiS2"...).

Phase 3: High-Throughput Inference

- **Featurization:** The backend converts these 50 strings into tensors using the CrabNet tokenizer.
- **Prediction:** The batch is sent to the GPU (mps).
- **Output:** The model returns a vector of predicted Formation Energies (ΔH_f) for each hypothetical compound.

Phase 4: Convex Hull Construction (The "Reaction")
This is the critical physics step. We use pymatgen.analysis.phase_diagram.

1. **Reference States:** The energy of the pure elements (Li, S) is set to 0 (by definition of

formation energy).
2. **Hull Calculation:** The algorithm identifies the subset of points that form the lower convex boundary.
    - *Mathematically:* It minimizes the energy $E(x) = (1-x)E_A + xE_B + \Delta H_{mixing}$.
3. **Stability Check:**
    - Compounds *on* the hull are **Stable**. (e.g., the model might find that "Li2S" lies on the hull).
    - Compounds *above* the hull are **Unstable**.

Phase 5: Response Generation
The backend returns the result to the EOH frontend.

- **Stable Products:** "Li2S" (Lithium Sulfide).
- **Reaction Energy:** The energy released per atom during the reaction (the depth of the hull).
- **Visualization Data:** The coordinates of the hull points, allowing the frontend to draw the phase diagram line.

6.2 Frontend-Backend Integration

To keep the system local and responsive, the backend should be exposed via a lightweight **FastAPI** server.

- **API Endpoint:** POST /predict_reaction
- **Latency:** The entire process (Sweep $\rightarrow$ Inference $\rightarrow$ Hull $\rightarrow$ JSON) should take < 200ms on an M1 chip, providing a "real-time" feel.

---

7. Future-Proofing: Integrating Pressure and Temperature

The user explicitly requested a roadmap for adding Pressure (P) and Temperature (T) variables. This represents a significant leap in complexity, moving from "Ground State (0 K) Prediction" to "Finite Temperature Thermodynamics."

7.1 The Physics of P and T

To predict stability at non-zero conditions, we must minimize Gibbs Free Energy (G) rather than Enthalpy (H).

$$G(P, T) = H(P, T) - T S(P, T)$$

$$G(P, T) = - T$$
This introduces two new terms that the current ML model does not predict: **Volume (V)** (for

the PV term) and **Entropy (S)**.

7.2 Implementation Strategy: The CALPHAD Bridge

The standard industry method for this is **CALPHAD** (Calculation of Phase Diagrams). We can implement a hybrid ML-CALPHAD approach using the **PyCalphad** library.[35]

Step 1: ML for Volume (Pressure)

- *Requirement:* To simulate high pressure, we must know the density of the phases.
- *Solution:* Train a secondary CrabNet model to predict **Volume per Atom** ($\text{\AA}^3/\text{atom}$).
- *Mechanism:* At Pressure P, add the term $P \times V_{pred}$ to the formation energy of each phase. Re-calculate the convex hull. This will correctly predict that denser phases (e.g., Diamond vs. Graphite) become stable at high pressure.[38]

Step 2: ML for Entropy (Temperature)

- *Vibrational Entropy (S_{vib}):* This drives solid-solid phase transitions. It is computationally expensive to calculate (requires Phonon Density of States).
- *Solution:* Train a tertiary ML model on the **JARVIS-DFT** or **Matbench Phonons** dataset to predict S_{vib} at 298K.[27]
- *Configurational Entropy (S_{conf}):* For solid solutions (alloys), PyCalphad can calculate the ideal mixing entropy analytically: $S_{mix} = -R [x \ln x + (1-x) \ln(1-x)]$.

Step 3: The Combined Pipeline

1. **Base Energy:** ML predicts E_{0K}.
2. **Pressure Correction:** ML predicts V; add PV.
3. **Temperature Correction:** ML predicts S_{vib}; subtract TS.
4. **PyCalphad:** Feed these temperature-dependent Free Energy functions ($G(T)$) into PyCalphad.
5. **Output:** PyCalphad minimizes the global energy and produces a Temperature-Composition Phase Diagram (showing melting points, eutectics, and solubility limits).[4]

---

8. Conclusion and Execution Roadmap

This report has defined a comprehensive architecture for a local, machine-learning-driven reaction prediction engine for the EOH platform. By synthesizing high-throughput DFT data with state-of-the-art attention-based deep learning, the system can accurately predict inorganic chemical reactivity without the need for trial-and-error.

Final Recommendations for the User:

1. **Dataset Decision:** Adopt a **Hybrid Strategy**. Use the **Materials Project** as the primary source for ground-truth stability but augment the training set with **OQMD** data to capture instability/negative examples. This provides the best balance of accuracy and robustness.
2. **Model Architecture:** Implement **CrabNet**. Its transformer-based design is the most capable of capturing the nuanced chemical interactions required for accurate energy prediction from composition alone.
3. **Hardware Strategy:** Leverage the **PyTorch MPS** backend. This is the most mature and stable path for accelerating deep learning on your local macOS environment.
4. **Future Expansion:** Design the API schema today to accept pressure and temperature parameters, but implement the backend logic for them in Phase 2 using **PyCalphad** and auxiliary Volume/Entropy ML models.

Execution Checklist

- [ ] **Setup:** Install Miniforge and PyTorch Nightly (MPS support).
- [ ] **Data:** Acquire API Key for Materials Project; script the bulk download of binary formation energies.
- [ ] **Train:** Train CrabNet on the MP+OQMD hybrid dataset; validate against a hold-out set (aim for MAE < 0.04 eV/atom).
- [ ] **Backend:** Build FastAPI wrapper with pymatgen Hull construction logic.
- [ ] **Integration:** Connect EOH frontend to localhost:8000.

This blueprint moves the EOH project from a conceptual visualization to a rigorously grounded computational chemistry tool, capable of providing instant, physics-informed insights into the nature of matter.

Works cited

1. Prediction and Classification of Formation Energies of Binary Compounds by Machine Learning: An Approach without Crystal Structure Information | ACS Omega - ACS Publications, accessed on December 5, 2025, https://pubs.acs.org/doi/10.1021/acsomega.1c01517
2. A Two-Step Machine Learning Method for Predicting the Formation Energy of Ternary Compounds - MDPI, accessed on December 5, 2025, https://www.mdpi.com/2079-3197/11/5/95
3. [2007.01988] Quantifying uncertainty in high-throughput density functional theory: a comparison of AFLOW, Materials Project, and OQMD - arXiv, accessed on December 5, 2025, https://arxiv.org/abs/2007.01988
4. Phase Diagrams (PDs) - Materials Project Documentation, accessed on December 5, 2025, https://docs.materialsproject.org/methodology/materials-methodology/thermodynamic-stability/phase-diagrams-pds
5. Phase diagram comic | HACKING MATERIALS, accessed on December 5, 2025, https://hackingmaterials.com/2014/05/08/pdcomic/

6. Create, plot, and analyze Phase Diagrams using the pymatgen codebase and Materials Project database - GitHub Gist, accessed on December 5, 2025, https://gist.github.com/computron/c0323115e92b48d0019d
7. Pretraining Strategies for Structure Agnostic Material Property Prediction - PMC, accessed on December 5, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10865364/
8. Compositionally restricted attention-based network for materials property predictions, accessed on December 5, 2025, https://www.researchgate.net/publication/351950098_Compositionally_restricted_attention-based_network_for_materials_property_predictions
9. Compositionally-Restricted Attention-Based Network for Materials Property Prediction | Theoretical and Computational Chemistry | ChemRxiv | Cambridge Open Engage, accessed on December 5, 2025, https://chemrxiv.org/engage/chemrxiv/article-details/60c756a40f50db3b10398109
10. Physical Encoding Improves OOD Performance in Deep Learning materials property prediction - arXiv, accessed on December 5, 2025, https://arxiv.org/html/2407.15214v1
11. OPTIMADE, an API for exchanging materials data - PMC - PubMed Central, accessed on December 5, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC8361091/
12. 6. Materials Data Access (Materials Project API Example) - YouTube, accessed on December 5, 2025, https://www.youtube.com/watch?v=Vuu7bNzmL8g
13. Python Materials Genomics (pymatgen) : A Robust, Open-Source Python Library for Materials Analysis. - CORE, accessed on December 5, 2025, https://core.ac.uk/download/pdf/78051478.pdf
14. Examples | Materials Project Documentation, accessed on December 5, 2025, https://docs.materialsproject.org/downloading-data/using-the-api/examples
15. Quantifying uncertainty in high-throughput density functional theory: A comparison of AFLOW, Materials Project, and OQMD | Request PDF - ResearchGate, accessed on December 5, 2025, https://www.researchgate.net/publication/372575802_Quantifying_uncertainty_in_high-throughput_density_functional_theory_A_comparison_of_AFLOW_Materials_Project_and_OQMD?_share=1
16. The Open Quantum Materials Database.markdown - GitHub Gist, accessed on December 5, 2025, https://gist.github.com/b9e747e7424d643dd441
17. Downloads - JARVIS-DB, accessed on December 5, 2025, https://jarvis-materials-design.github.io/dbdocs/thedownloads/
18. JARVIS-DFT | NIST - National Institute of Standards and Technology, accessed on December 5, 2025, https://www.nist.gov/programs-projects/jarvis-dft
19. Inorganic Phase Equilibrium Data | NIST - National Institute of Standards and Technology, accessed on December 5, 2025, https://www.nist.gov/programs-projects/inorganic-phase-equilibrium-data
20. List of useful databases | Chemistry Library - University of Cambridge, accessed on December 5, 2025, https://library.ch.cam.ac.uk/list-useful-databases

21. Featurizers — deepchem 2.8.1.dev documentation, accessed on December 5, 2025, https://deepchem.readthedocs.io/en/latest/api_reference/featurizers.html
22. Machine-Learning Prediction of Curie Temperature from Chemical Compositions of Ferromagnetic Materials - PMC - PubMed Central, accessed on December 5, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11350867/
23. Machine-Learning Prediction of Curie Temperature from Chemical Compositions of Ferromagnetic Materials - ACS Publications, accessed on December 5, 2025, https://pubs.acs.org/doi/10.1021/acs.jcim.4c00947
24. How to Visualize Machine Learning Models with Python - DataCamp, accessed on December 5, 2025, https://www.datacamp.com/tutorial/visualize-machine-learning-models
25. [2309.04482] Addressing the Accuracy-Cost Tradeoff in Material Property Prediction: A Teacher-Student Strategy - arXiv, accessed on December 5, 2025, https://arxiv.org/abs/2309.04482
26. Benchmark results. MAE scores of Roost, CrabNet, onehot encoded CrabNet... | Download Scientific Diagram - ResearchGate, accessed on December 5, 2025, https://www.researchgate.net/figure/Benchmark-results-MAE-scores-of-Roost-CrabNet-onehot-encoded-CrabNet-HotCrab-and_fig1_351950098
27. matbench_v0.1 matbench_expt_gap - MatBench, accessed on December 5, 2025, https://matbench.materialsproject.org/Leaderboards%20Per-Task/matbench_v0.1_matbench_expt_gap/
28. How to optimize data science packages in Python for Apple Silicon M1/M2 | by John Medina, accessed on December 5, 2025, https://medium.com/@i.johnmedina/how-to-optimize-data-science-packages-in-python-for-apple-silicon-m1-m2-d659d53ecea8
29. Installing scipy and scikit-learn on apple m1 - python - Stack Overflow, accessed on December 5, 2025, https://stackoverflow.com/questions/68620927/installing-scipy-and-scikit-learn-on-apple-m1
30. MLX a machine learning framework for Apple Silicon - Macs in Chemistry, accessed on December 5, 2025, https://macinchem.org/2023/12/08/mlx-a-machine-learning-framework-for-apple-silicon/
31. MLX - Apple Open Source, accessed on December 5, 2025, https://opensource.apple.com/projects/mlx
32. ml-explore/mlx: MLX: An array framework for Apple silicon - GitHub, accessed on December 5, 2025, https://github.com/ml-explore/mlx
33. XGBoost GPU Support — xgboost 3.1.1 documentation, accessed on December 5, 2025, https://xgboost.readthedocs.io/en/stable/gpu/
34. Apple Silicon Compatibility · Issue #6408 · dmlc/xgboost - GitHub, accessed on December 5, 2025, https://github.com/dmlc/xgboost/issues/6408
35. pycalphad: Computational Thermodynamics - Phases Research Lab, accessed on December 5, 2025, https://phaseslab.com/pycalphad
36. pycalphad/pycalphad: CALPHAD tools for designing thermodynamic models,

calculating phase diagrams and investigating phase equilibria. - GitHub, accessed on December 5, 2025, https://github.com/pycalphad/pycalphad
37. PyCALPHAD, accessed on December 5, 2025, https://pycalphad.org/docs
38. Machine learning-based modeling of high-pressure phase diagrams: Anomalous melting of Rb | The Journal of Chemical Physics | AIP Publishing, accessed on December 5, 2025, https://pubs.aip.org/aip/jcp/article/157/1/014502/2841551/Machine-learning-based-modeling-of-high-pressure