Project Nova:
AI-Driven Platform for Autonomous Materials Discovery & Additive Manufacturing Systems

Executive Summary

The following document outlines a comprehensive technology specification for an **AI-driven, fully automated materials discovery platform** integrated with Desktop Metal additive manufacturing systems. This platform combines advanced software (notably large language model agents and multi-modal AI) with a state-of-the-art autonomous manufacturing lab to **design, fabricate, and evaluate new materials without human intervention**. Key features include real-time data exchange with Desktop Metal printers (Shop System™, Production System P1, X25 Pro, X160 Pro), closed-loop feedback for print optimization, and a roadmap toward fully robotic sample handling (printing, depowdering, sintering, and testing). The system architecture is **generalizable across diverse R&D domains** – from rare-earth-free permanent magnets to solid-state battery electrolytes – enabling accelerated innovation through autonomous experimentation. Recent advances in AI for chemistry and robotics are leveraged (e.g. LLM-based design frameworks, vision-guided robots), positioning this platform at the forefront of self-driving labs. Ultimately, the specification aims to inform R&D and engineering stakeholders on how to build the **world's first truly autonomous materials discovery lab**, with clarity on the architecture, components, AI capabilities, data workflows, integration strategy, development roadmap, and representative use cases.

System Architecture

Figure: High-level system architecture of the AI-driven autonomous lab. The platform's architecture centers on an **AI Orchestrator** (an autonomous agent powered by LLMs) that coordinates all subsystems – additive manufacturing hardware, robotics, sensors, and data infrastructure. It follows a closed-loop "self-driving" lab paradigm where **experimental design, execution, and analysis are tightly integrated** llnl.govllnl.gov. The major architectural layers include:

- **AI Reasoning Layer:** The cognitive engine (LLM-based) interprets goals, generates experimental plans, and makes decisions in real time. It interfaces with tool APIs and databases to ground its reasoning in both prior knowledge and live data (for example, querying materials databases or reading sensor feeds).

- **Hardware & Robotics Layer:** A network of Desktop Metal **3D printers** (Shop, P1, X25 Pro, X160 Pro) for sample fabrication is augmented by **robotic systems** (robotic arms, gantries, or conveyors) that handle physical tasks – moving build plates, depowdering parts, loading/unloading a **sintering furnace**, etc. Ancillary equipment like **depowdering stations, furnaces, and testing instruments** are all connected as nodes in the automated workflow.

- **Data & Sensing Layer:** A unified data pipeline connects all devices and processes. **IoT sensors and cameras** on the printers and lab equipment stream real-time status and environmental data (e.g. chamber temperatures, print progress, video of the print bed) into the system desktopmetal.com. All experimental data – print parameters, in situ sensor readings, post-process measurements – are logged in a central **Materials Data Repository** for analysis and machine learning.

- **Integration & Interface Layer:** Standardized interfaces (APIs, possibly via Desktop Metal's Live Suite and ROS for robotics) enable seamless communication. The AI Orchestrator uses these interfaces to dispatch print jobs, adjust parameters, control robots, and retrieve results. A **user interface** sits above, allowing human researchers to set objectives or review results at a high level, while the AI handles low-level execution.

This architecture ensures **seamless coordination** between software intelligence and laboratory hardware. By design, it supports iterative experimentation: the AI agent can plan a material composition, command its fabrication and processing, receive outcome data, then **learn and refine subsequent experiments autonomously** llnl.govllnl.gov. This modular, extensible architecture will allow new machines or analytical instruments to be added with minimal reconfiguration, fostering adaptability to evolving R&D needs.

Components

Hardware Components

- **Desktop Metal Additive Manufacturing Systems:** The platform integrates multiple 3D printers from Desktop Metal's portfolio, each suited for particular tasks:
  - *Shop System*™: A binder jet metal printer ideal for batch production of small-to-medium parts. It will be used for rapid prototyping of material coupons and combinatorial samples. The Shop's job configuration (layer thickness, binder saturation, etc.) is programmatically controlled via the AI agent.
  - *Production System P1*: A single-pass inkjet (binder jetting) printer with higher throughput, used for screening a large number of compositions quickly. The P1's

high speed allows the AI to iterate faster on promising material candidates.

  - *X25 Pro and X160 Pro*: Large-format binder jet printers (originating from ExOne's technology) for scaling up sample size or producing arrays of samples in one build. These systems can fabricate numerous test specimens in parallel, which is crucial for experiments requiring statistical validation. They also support a wide range of materials – various metals, ceramics, even multi-material builds – providing flexibility in materials research desktopmetal.com.

  - All printers are equipped with **onboard sensors** (e.g. powder bed cameras, temperature and humidity sensors, actuator monitors). Through Desktop Metal's software (Live Suite's Live Monitor), the platform can pull telemetry data and receive alerts in real time desktopmetal.com. Print job APIs (or SDKs) allow the AI to start prints, adjust parameters, and pause or abort jobs if anomalies are detected.

- **Sintering Furnace (and Ancillary Post-Processing Equipment):** Since the metal and ceramic parts printed via binder jet are "green" (unsintered), an **automated furnace** is integrated for debinding and sintering cycles. For example, the Desktop Metal Furnace (e.g. the Shop System furnace or a custom high-temperature furnace) can be controlled via programmatic recipes. The AI orchestrator sends a sintering profile (temperature ramp, hold times, atmosphere control) tailored to each material. A **furnace status interface** provides live feedback on temperature curves and completion times, closing the loop on the print-to-sinter process. The roadmap includes outfitting the furnace with an automated door and perhaps a small robotic mechanism or conveyor so that parts can be loaded/unloaded without human help.

- **Depowdering & Cleaning Station:** Printed parts need to be freed from loose powder. The platform will include a **sealed depowdering station** – essentially a chamber with brushes, air jets or ultrasonic vibration to remove excess powder. In early implementation, an operator might place the build box in the station, but the goal is full automation: a robot inserts the printed build into the depowdering unit, and after an automated cycle, retrieves the cleaned parts. The station is equipped with sensors (cameras to verify powder removal, load cells to weigh recovered powder) and connects to the data system to log the amount of powder recycled or any anomalies (e.g. a part that broke during cleaning).

- **Robotic Handling Systems:** Multiple robotic components ensure physical automation:

  - *Robotic Arm(s)*: e.g. a 6-axis robotic arm on a rail that can reach the printers, furnace, and depowdering station. It is responsible for transferring build plates or parts between each process step. For instance, once a print job is done, the arm can pick up the build box from the printer, move it to the depowder station, then after cleaning, transfer the parts to a sintering tray and into the furnace.

High-precision fiducial markers and computer vision will enable the robot to align accurately with equipment. The robot operates under a coordination system (likely ROS) that takes high-level commands from the AI (like “retrieve print from Printer 1”) and breaks them into motions (open door, pick object, etc.).

  - *Automated Guided Vehicle (AGV) or Conveyor*: Optionally, for labs with larger footprints or multiple stations, small AGVs or conveyor belts can transport materials between fixed stations. This is particularly useful if the furnace is not co-located next to the printer or if multiple printer cells feed into a central sintering area.

  - *Tool Changers and Effectors*: The robotic arm may use different end-effectors – e.g. a gripper for picking up build plates, a vacuum nozzle for handling powder containers, or even a specialized magnetized gripper for ferromagnetic parts. Including a tool-changing system extends the robot’s versatility in handling various material forms and containers.

- **Sensors and Instruments:**

  - *Vision Systems*: Cameras are placed for multiple purposes – inside printers (to monitor layer deposition), inside the depowdering chamber (to ensure thorough cleaning), and on robotic arms (for situational awareness and part localization). Computer vision (potentially powered by an AI vision model) can detect print defects (layer shifting, delamination) or verify successful picks and placements by the robot.

  - *Material Characterization Tools*: To truly complete the discovery loop, the platform can integrate certain analytic instruments. Examples include a **microscope (optical or SEM)** to examine microstructure of printed/sintered samples, a **hardness tester** or **indenter** (as used in LLNL’s APEX for mechanical testing llnl.gov), a **vibrating sample magnetometer (VSM)** for measuring magnetic properties, or an **electrochemical impedance spectroscopy setup** for battery electrolyte evaluation. In early stages, these might be offline measurements done by scientists, but later they can be automated (e.g. a robot could place a sample in a magnetometer and initiate a test). Sensor readings from these devices are fed back into the central database.

  - *Environmental Sensors*: For safety and process control, sensors for temperature, humidity, oxygen level (especially if working with reactive materials in glovebox or inert atmosphere) are included. The system monitors these to ensure safe operating conditions (for example, ensuring the build chamber is dry if printing moisture-sensitive powders, or that no inert gas leak occurred in a glovebox).

- **Computing Infrastructure (Hardware):** The AI orchestrator and data processing pipelines will run on dedicated computing resources. This includes servers or cloud instances with **GPUs** (for machine learning tasks and vision processing) and possibly an on-premises **edge computer** that interfaces with lab hardware with low latency (running the ROS robot controller or real-time monitor for printers). Data storage (a database or data lake) is also provisioned, potentially with high-speed SSD storage for fast logging of sensor data. For simulation tasks (e.g. running physics simulations like sintering prediction or DFT calculations for materials), HPC resources or cloud compute can be leveraged asynchronously.

Software Components

- **AI Orchestrator (LLM Agent Framework):** At the heart is a software agent built around a **Large Language Model** (such as GPT-4 or a domain-specialized LLM). This agent operates as the “brain” of the lab, capable of understanding objectives, breaking them into tasks, and orchestrating tools. It uses a **self-reflective, iterative reasoning loop** inspired by frameworks like *LLMatDesign*, which treat materials design as a closed-loop dialogue of hypotheses and experiments. The AI orchestrator is implemented with an agent loop that can: 1) **Plan** an experiment (propose a material composition or process adjustment), 2) **Execute** by interfacing with hardware (e.g. sending print instructions), 3) **Analyze** the outcome (compare target vs measured results), and 4) **Learn/Reflect** to improve the next iteration. This loop continues autonomously until termination criteria are met (e.g. a material meeting the target performance is found, or a maximum number of iterations reached). The LLM is augmented with domain knowledge via prompt engineering and tool use – for example, it can call a materials property predictor or search literature for guidance, then integrate that into its decision-making.

- **Materials Knowledge Database:** A central database stores all experimental data and relevant domain knowledge. This includes a catalog of materials compositions tried, processing parameters used, and measured outcomes (properties, microstructures, etc.), as well as external data like materials databases or prior literature data. The AI uses this database both as a memory (to avoid repeating failed experiments and to analyze trends) and as training data for surrogate models. Over time, as the number of experiments grows, this repository becomes extremely valuable; it can be mined to train machine learning models (for example, a model predicting magnetic coercivity from composition and microstructure, or a model predicting optimal sintering temperature for density). The database also contains **digital twin** information of the hardware – e.g. calibration data, machine-specific parameters – enabling the AI to account for machine variation in its plans.

- **Machine Interfaces & APIs:** Software adapters connect the AI orchestrator to each piece of hardware:

- For Desktop Metal printers, the **Live Suite API** (or a custom interface) is used. Initially, this may involve using **Live Studio/Live Build** software in automated mode: the AI provides a CAD or build file and material profile, which the software slices and sends to the printer. In later integration, the AI might directly control slicing parameters or call a **Live Build optimization** tool to arrange many samples in one build efficiently. **Live Monitor** is used to pull real-time status and sensor data from the printers desktopmetal.com.

- The **Robot Control** is handled via a robotics middleware (e.g. **ROS2 (Robot Operating System)**). The AI sends high-level commands (in natural or structured language) to a planning module, which uses motion planning and execution to drive the robot. Notably, recent research (like VoxPoser and others) demonstrates that LLMs can output robot motion plans by interacting with vision models and generating intermediate representations openreview.net. Our system may incorporate such techniques: for instance, the AI could request a pick-and-place by describing the target object, and a backend module (with computer vision and motion primitives) will compute the trajectory. The low-level motor control is handled by robot drivers, while the AI monitors success/failure signals (possibly using **GPT-4 Vision** capabilities to verify via camera images that, say, a part was successfully picked up).

- **Laboratory Instruments** (furnaces, sensors, measurement devices) are integrated via appropriate protocols (industrial automation protocols or instrument-specific APIs). For example, the furnace might be controlled via Modbus/OPC UA or simply a Python library provided by the manufacturer. The AI can start a sintering cycle and subscribe to event callbacks (like "cycle complete" or temperature reached). Measurement devices, if computer-controlled (like a programmable multimeter or impedance analyzer), can be scripted: the AI triggers a measurement and then either directly reads the result or receives it via the database.

- **Surrogate Modeling and Simulation Tools:** To accelerate decision-making, the platform includes models that can predict outcomes without needing a full physical experiment each time. Examples:

  - **Live Sinter™ Simulation** desktopmetal.com: Desktop Metal provides a sintering simulation tool to predict shrinkage and deformation. The AI can use this tool on a candidate geometry before printing, to foresee and correct distortion issues or adjust the model (a form of in silico feedback).

  - **Materials Property Predictors:** In line with LLMatDesign usage of ML models as stand-ins for costly calculations themoonlight.io, the platform will host AI/ML models to predict material properties (e.g., a model that estimates a magnet's coercivity from its composition and density, or a model predicting ionic conductivity of a ceramic from its chemistry). These predictors can be pre-trained on literature data or gradually improved with the experimental data collected. The AI orchestrator queries these models when evaluating a proposed material **before committing to print**, to filter out poor candidates or to choose the most promising experiments.

  - **Physics-based Simulators:** Where applicable, the system can incorporate first-principles or physics simulations in the loop. For instance, density functional theory (DFT) calculations could be queued (perhaps running on an HPC cluster) to compute formation energies or electronic structures of newly proposed materials. While too slow for every iteration, these can be used sparingly for high-value candidates or to validate the AI's suggestions in parallel to the physical experimentation.

- **User Interface and Monitoring Dashboard:** Although the platform is autonomous, a web-based dashboard allows human researchers to interact with it. They can input high-level objectives (e.g. "Find a magnet alloy with coercivity ≥ 1000 kA/m at 150°C" or "Optimize the sintering profile for maximum density without grain growth") in natural language. The UI displays the experiment queue, real-time status of each device, and the results as they come in (with charts for properties over iterations, etc.). It also provides transparency into the AI's reasoning – showing, for example, the hypotheses it generated and the self-reflections after each experiment. This builds trust and helps experts guide the AI if needed. Additionally, safety controls are accessible: humans can pause or stop the automation via the UI in case of emergencies.

- **Self-Optimization and Meta-Learning Module:** As part of the software, there is a component dedicated to optimizing the platform's own performance. This includes routines for **print parameter optimization** and calibration. For example, the system will autonomously tune printer settings (binder saturation, layer curing times, recoater speed) to improve print quality for new materials – effectively learning the best process parameters as it goes. It can perform Design of Experiments or use Bayesian optimization to hone in on settings that yield high-density, defect-free parts, all recorded in the database for future reference. Similarly, for robotic actions, the system can learn from failures (if a pick fails, adjust approach) and maintain calibration data (using fiducials and sensor feedback to auto-correct robot positioning over time).

In sum, these components – from smart AI algorithms to robust hardware integration – form a cohesive ecosystem. The **software orchestrates the hardware** to perform complex research tasks, while the hardware provides the means to execute and gather data, which in turn feeds back to improve the software's models. This synergy is what enables the platform to operate as a "self-driving lab."

AI Capabilities and Intelligence Framework

The platform leverages cutting-edge AI capabilities to drive materials discovery in an **autonomous, intelligent manner**. Key aspects of the AI system include:

- **Large Language Model (LLM) as a Central Reasoning Engine:** At its core, the AI orchestrator is built on a powerful LLM (such as GPT-4 or a specialized scientific LLM). This gives the system a form of **generalist intelligence** – it can parse human language inputs, assimilate domain knowledge, and reason through complex tasks. Unlike traditional materials discovery approaches that require extensive numeric modeling or costly data training, an LLM-based approach can utilize contextual knowledge from scientific literature and adapt on the fly. *For example:* if the user's goal is to develop a rare-earth-free magnet, the LLM can recall (from its training or provided references) what alloy systems are promising, known issues (e.g. "Mn-Bi compounds have high coercivity at high temperature" or "Ce-substitution in Nd-Fe-B reduces performance"), and suggest avenues to explore. This **knowledge-driven reasoning** jump-starts the experimental planning in ways purely data-driven methods might miss.

- **Interpreting Natural Language & Multi-Modal Inputs:** The LLM allows the system to understand **unstructured instructions** and also to generate human-readable explanations of its plans. Researchers can give high-level instructions in English, and the AI will translate that into a series of experiments. Conversely, the AI can explain why it's trying a certain composition or processing change in understandable terms (a boon for interpretability and trust). Moreover, the AI is extended to be **multi-modal** – it can process not just text, but also image and numerical data. For instance, the orchestrator might analyze a microscope image of a material's microstructure: using a vision model or the multi-modal extension of GPT-4, it could describe the grain morphology or detect the presence of cracks or secondary phases. It can also read plots or charts (e.g. a stress-strain curve or magnetic hysteresis loop) to extract quantitative metrics. This multi-modal reasoning enables a richer feedback loop; the AI isn't blinded to anything that isn't textual – it can truly ingest the full breadth of experimental observations.

- **Autonomous Planning and Tool Use:** The AI agent operates with a degree of autonomy in planning experiments. It uses an approach akin to a **planning algorithm combined with chain-of-thought reasoning**. Concretely, the LLM internally breaks tasks into sub-tasks and decides on actions, often using an approach like: *Goal → Thought → Action → Observation → Thought → …*, similar to how an expert human might approach the problem. Crucially, the AI is **augmented with tools**. It can call external functions or models via a predefined API (this could be implemented via an agent framework where the LLM's output is parsed for commands). For example, if it needs to evaluate a candidate material's properties, it can call the property predictor tool or a quick DFT simulation. If it needs to plan a robot motion, it can invoke a motion planning routine or even output code (Python/ROS scripts) that get executed – leveraging the **code-writing capability of LLMs** to control hardware openreview.net. This paradigm was demonstrated by *VoxPoser*, where GPT-4 generated code to compose 3D value maps for robotic trajectories openreview.net. In our platform, we envisage the LLM writing small snippets (or selecting from a library of scripts) to, say, adjust a printer setting or analyze a dataset. The AI's ability to **integrate symbolic reasoning with low-level control** is a game-changer for automating lab tasks that traditionally required separate control logic.

- **Self-Reflection and Learning from Experience:** The AI is designed with a *self-reflective loop* to improve its performance over time. Inspired by frameworks like LLMatDesign, after each experiment the AI not only records the outcome but also **analyzes the success or failure of its strategy** themoonlight.io. For example, if adding a certain element to an alloy decreased the performance, the AI will reflect (via an internal prompt mechanism) on why that might have happened ("perhaps the addition formed a brittle phase that reduced magnetic alignment") and log this reasoning. This reflection is then used to adjust subsequent proposals themoonlight.io. The LLM essentially engages in a dialogue with itself, refining hypotheses – much like a researcher writing in a lab notebook and thinking about what to try next. This approach allows rapid adaptation to new tasks in a zero-shot or few-shot manner arxiv.org. In practice, this means the platform doesn't require thousands of prior experiments to begin making good decisions; it can start intelligent exploration on day one, and only gets better as it gathers data.

- **Incorporation of Domain Knowledge and Constraints:** The AI's reasoning is bounded and guided by scientific domain knowledge. Hard physical constraints (like "sample must sinter below 1400°C to avoid evaporating element X" or "the printer's resolution is 50 µm so features smaller than that may not form") are encoded either in prompt hints or enforced by rule-checkers that post-process the AI's suggestions. The LLM is also fed with **relevant literature and patents** (via vector databases or retrieval-augmented generation) so that it can cite known results and not rediscover known failures. For instance, if an entire class of alloys was already found to be unsuitable, the AI will recognize that from the knowledge base and avoid wasting time on them, focusing instead on less-explored combinations. This makes the search **more efficient and informed** than a blind algorithmic search.

- **Advanced Scientific Reasoning Capabilities:** We leverage studies demonstrating that modern LLMs possess surprisingly strong scientific reasoning in chemistry and beyond. According to recent benchmarks (*ChemBench*), the best LLMs can outperform expert chemists on many knowledge-based questions, while still struggling on certain fundamental tasks nature.com. The platform capitalizes on these strengths – e.g., the LLM can answer complex chemistry questions ("What elements could improve Curie temperature without adding rare-earths?") by synthesizing knowledge from its training data. At the same time, we mitigate the weaknesses: the AI's plans are always cross-validated either by simulation or a small-scale test to guard against any overconfident hallucinations from the LLM nature.com. In essence, the LLM is a very knowledgeable co-pilot, but the system architecture provides a safety net (via real experiments and calculations) to ensure that only physically-valid outcomes are pursued.

- **Multimodal Robotic Reasoning:** When it comes to controlling physical hardware, the AI employs a combination of high-level reasoning and low-level feedback control. For example, consider the task of depowdering a part. The AI might issue a command like “rotate part and brush off remaining powder” – behind the scenes, this is resolved by a vision-based agent that uses visual feedback to see where powder remains and guides the robot accordingly. Research like the recent **ELLMER robot** (Embodied LLM) shows that combining an LLM with real-time visual and force feedback enables complex multi-step tasks to be handled in unstructured environments azorobotics.com. Our platform will implement similar capabilities: the AI can adapt if, say, a part is not exactly where expected or a task fails. It retrieves context (through sensors) and adjusts actions dynamically, rather than following a rigid script. This makes the automation robust to surprises (e.g., a printed part is slightly warped so the robot’s usual grip strategy needs altering – the force sensor detects slippage and the AI adjusts grip approach on the fly).

- **GPT-4 Scientific Insight Integration:** By harnessing GPT-4’s broad scientific knowledge (often described in literature as showing *sparks of AGI* in scientific domains nature.com), the platform can do things like generate hypotheses that merge knowledge from different fields. For instance, GPT-4 was noted to not only solve exam questions but also suggest experimental plans when given the right tools nature.com. Here, with direct access to experiment execution, the AI can truly “think and do” science. It could propose an unconventional approach (perhaps inspired by a distant analogy in biology or physics) that a narrower algorithm might never consider. This creativity, guided by real-world testing, could lead to serendipitous discoveries.

In summary, the AI capabilities of this platform revolve around **an LLM-centric autonomous agent** that is knowledgeable, tool-using, reflective, and adaptive. It integrates multi-modal feedback (text, image, sensor data) and drives the entire materials R&D cycle. By building on recent advances like LLMatDesign’s iterative self-improvement themoonlight.io, VoxPoser’s language-to-robot action translation openreview.net, and ChemBench’s insights on LLM chemical knowledge nature.com, the platform ensures that its AI is not a black-box oracle but an interactive, improving partner in experimentation. The result is a system that can **reason like a scientist and act like an engineer**, continuously learning to become a better researcher over time.

Workflow & Data Pipeline

The platform operates through a **closed-loop experimental workflow** that continuously cycles between hypothesis generation, experiment execution, and data analysis. This section describes the end-to-end workflow and how data flows through the system at each stage:

1. **Experiment Planning (Digital Stage):** The process begins with the AI Orchestrator formulating an experiment. This could be initiated by a human-defined goal (e.g., "discover a magnet alloy with <5% rare-earth content and coercivity > 1000 kA/m") or by the AI's own strategy (e.g., exploring around a promising composition found in a previous run). The LLM agent considers the current knowledge state (previous results, known literature, model predictions) and decides on a **hypothesis and experiment design**. For example, it might propose: *"Try adding 2% cobalt to the Fe-Ni alloy to improve its Curie temperature."* Along with the hypothesis, it plans the fabrication process: selecting which printer to use, what geometry the sample should be, and any specific print parameter tweaks. The output of this stage is a *digital experiment plan* – essentially a set of instructions and expected outcomes.

2. **Sample Fabrication (Physical Stage):** The plan is executed in the lab. The AI sends the design (CAD or build file) and job parameters to a Desktop Metal printer via the integration API. The **printer begins the 3D printing process**, laying down material layer by layer. Throughout the print, **data is collected in real time**: the Live Monitor feed provides layer images and sensor logs (e.g., binder flow rate, bed temperature). This data streams into the central database and is also monitored by the AI for any anomalies. If a severe deviation is detected (e.g., a layer printing incorrectly or a machine error), the AI can decide to pause or abort the print, flagging the issue for analysis. Otherwise, upon completion, the printer signals the job is done and hands off the printed part (often still in a powder bed) to the next step.

3. **Post-Processing (Physical Stage):** Once printing is complete, **robotic handling** takes over for post-processing:

    - The robotic arm retrieves the build box and transports it to the **depowdering station**. In the depowdering step, which may involve automated brushing or blasting, sensors (like an internal camera) watch the process. The system verifies that the part is successfully cleaned (comparing images of the part before and after, or ensuring the weight of recovered loose powder meets expectations).

    - Next, the cleaned "green" part is moved to the **sintering furnace**. The AI selects a sintering program (time-temperature profile) appropriate for the material – this could be one recommended by prior simulations or adjusted based on past experiments. The furnace is then controlled automatically to run that cycle. During sintering, **thermal data** (temperature vs. time, etc.) is logged. If the furnace has monitoring like shrinkage measurement (some advanced systems use a camera to watch dimensional changes), those data are captured as well.

    - Depending on the experiment, additional post-processing might be needed. For magnets, for example, the sample may need to be **magnetized** by exposing it to a strong magnetic field after sintering. If so, the workflow includes a station (like an electromagnet setup) where the robot places the sample for magnetization.

Similarly, if the part needs **polishing** (as in the APEX alloy case for microstructure prep llnl.gov), a robotic polishing step could be inserted.

4. **Characterization & Testing (Physical Stage):** After the sample is fully prepared, the platform performs **automated characterization tests** to measure the target properties:
   - For the magnet example, the system would measure magnetic properties such as coercivity, remanence, and magnetization. This could involve placing the sample in a vibrating sample magnetometer (VSM) or a custom test rig with Hall sensors. The AI triggers the measurement and retrieves the results (e.g. coercivity value, BH curve data).
   - For a battery electrolyte, the platform would perform ionic conductivity tests. For instance, it might sandwich the printed electrolyte pellet between two sodium metal electrodes in a test cell and run electrochemical impedance spectroscopy (EIS). The control software would sweep frequency and measure impedance, and from that the AI can calculate ionic conductivity.
   - If multiple properties need evaluation (hardness, electrical conductivity, etc.), the sample might go through several instruments. The robotics can route the sample through a sequence of stations if needed.
   - Additionally, **structural or compositional analysis** may be performed. The platform could take a microscope image (automatically focusing and capturing a micrograph of the sample's cross-section) or even use an X-ray fluorescence (XRF) or diffraction (XRD) instrument to verify phase composition. All these tests generate data that feed back into the system.
5. **Data Aggregation and Logging:** All results from fabrication and testing are aggregated in the **Materials Data Repository** under an entry for that experiment. A single experiment record might include: the intended composition and process, the printer logs, the sintering profile, and the measured outcomes (density, microstructure images, property values). Metadata like timestamps, equipment used, and any anomalies are also stored. The data pipeline ensures that this information is **indexed and accessible** – e.g., the AI can query "find all experiments where coercivity was above 800 kA/m" or "retrieve the micrograph from experiment #37".
6. **Analysis and Feedback (Digital Stage):** Now the AI Orchestrator processes the outcome relative to the hypothesis:
   - First, it checks the **target criteria**. Was the goal achieved (yes/no)? How close was it? Perhaps the result is promising (e.g. coercivity improved by 20% but still short of the goal).

- The AI then performs a **post-mortem analysis**. It might compare the result to predictions: if the surrogate model predicted a higher performance than observed, why the discrepancy? It could hypothesize reasons, such as "the addition of Co improved Curie temperature but maybe caused brittleness that lowered overall magnetization." It will examine any ancillary data (microstructure images, etc.) to support or refute these hypotheses.

- Crucially, the AI **updates its knowledge**. The new data point can be used to refine the surrogate models (for instance, incorporate it into a training set for the property predictor to make it more accurate for that composition space). If the platform uses Bayesian optimization or another active learning strategy, this result is used to update the model posterior or acquisition function for proposing the next experiment.

- The self-reflection mechanism kicks in here: the AI records what it *thought* would happen vs. what actually happened and adjusts its internal plan accordingly themoonlight.io. If the experiment succeeded, it learns what success looks like; if it failed, it learns to avoid that path or to understand the failure mode.

7. **Next Iteration Decision:** Based on the analysis, the AI decides what to do next. If the target hasn't been met, it will propose a new experiment – this forms a **closed-loop iterative cycle** llnl.gov. For example, "Co addition helped, but magnetization dropped; next, try adding a grain boundary element like Bismuth to isolate grains," referencing known strategies ameslab.gov. The entire loop (planning to analysis) then repeats for the new hypothesis. If the target *has* been achieved or the experiment set reaches a stopping criterion, the AI can conclude the campaign and output the best found solution and insights.

8. **Parallelization and Campaign Management:** The above describes a single experimental thread, but the system can manage **multiple experiments in parallel**. With multiple printers and perhaps multiple test devices, the AI might run several campaigns simultaneously (for instance, one focusing on magnets, another on electrolytes, each with its own loop). It schedules jobs on available printers, prioritizes experiments by expected value of information, and ensures that the data from all threads is properly logged without confusion. The data pipeline is designed to tag each data stream with experiment IDs to keep them separate. Moreover, parallel experiments can inform each other if relevant – the AI can transfer learnings (maybe a technique learned in one campaign is applicable to another).

9. **Human-in-the-Loop (optional):** While the platform is autonomous, at any point humans can intervene or collaborate. For instance, a researcher might see an interesting unexpected result via the dashboard and ask the AI to investigate it more deeply ("focus on why sample #5 had an anomalous microstructure"). The AI can accommodate such requests, essentially altering its planned sequence to satisfy the query (this might involve

doing additional characterization on that sample or running a slight variation experiment to isolate a variable). This flexible data pipeline and workflow management allow on-the-fly adjustments while keeping the overall automation intact.

Throughout this workflow, **data integrity and feedback** are paramount. Every stage provides data that the next stage or the AI analysis can consume, forming a **continuous feedback loop** that accelerates learning llnl.gov. The platform effectively implements the scientific method in an automated way: form a hypothesis, test it, evaluate results, then refine the hypothesis – all at machine speed. By capturing the full lineage of each sample (from digital design to measured performance), the system can trace causes of success or failure with high confidence, enabling rapid optimization of materials and processes.

Moreover, the data pipeline is built to be scalable and **shareable**. In a multi-user or consortium setting, experiment data and AI models could be shared in the cloud, contributing to a broader Materials Acceleration Platform (as LLNL's MAP framework envisions combining models and experiments) llnl.govllnl.gov. This means the discoveries made by the platform not only yield a candidate material but also enrich the collective knowledge base, which in turn improves future experiment planning – a virtuous cycle of acceleration.

Integration Strategy with Desktop Metal Systems

To achieve seamless integration with Desktop Metal's additive manufacturing hardware, a clear strategy is devised that addresses communication, control, and feedback between the AI platform and the manufacturing equipment. The integration will be implemented in **phases**, ensuring that initial capabilities are established early (data connectivity, remote monitoring) and then progressively expanded (active control, full automation). Key elements of the strategy include:

Phase 1: Data Connectivity and Monitoring

Goal: Enable the AI platform to **monitor Desktop Metal printers and retrieve process data** in real-time, and to log print outcomes, without yet taking direct control of machine operation.

- **Utilize Desktop Metal's Live Suite APIs:** Desktop Metal provides software tools (Live Monitor, Live Studio, etc.) that are designed for user monitoring and job management desktopmetal.com. In this phase, we leverage any available external interfaces of these tools. For example, Live Monitor can stream real-time status – we will connect to it via network API or a webhook system if provided. This allows the AI to read information such as print progress (% completion, current layer), current machine parameters, and any error codes. If an official API is not available, we will use a **middleware service** that can pull data from the printer's internal logs or even IoT sensors attached to the printer (essentially an adapter that translates the printer's data into a REST or MQTT feed for the AI platform).

- **Implement a Data Ingestion Pipeline:** The integration will include a **connector service** running on a local server or cloud that continuously pulls data from the printers and pushes it to the platform's database. For reliability, this service handles reconnection, buffering (in case of network hiccups), and time-stamping of data. All data from the printer – build ID, layer count, temperatures, etc. – are stored and linked to the experiment context so the AI can later analyze it.

- **One-Way Control (Submit Jobs via Official Workflow):** In Phase 1, we assume print jobs are still initiated through Desktop Metal's standard software (for safety and to not void warranties). However, we integrate by having the AI **programmatically interface with job submission tools**. For instance, the AI can generate a CAD model or use a template file, then call a script that drives Desktop Metal's Live Studio (which might have a command-line or API to load a model and queue a print). Alternatively, the AI outputs an STL/BST file and a human or a simple script feeds it into the printer software. The emphasis is on minimal hacking – using the vendor-provided workflow but automated. This ensures that slicing and machine calibration settings are correctly applied by the official software, reducing risk. The integration at this stage is akin to a "headless user" of the printer software, not a firmware-level control.

- **Feedback Loop Establishment:** At this stage, if the AI wants to adjust something mid-print (which is rare in binder jetting – generally prints just run once configured), it would likely plan to do so between prints rather than during one. The key integration is getting **post-job feedback**: once a print completes, the AI is automatically notified (either by polling job status or via a callback). It then can retrieve the outcome data, such as a confirmation that the print finished successfully or if the machine reported a fault (e.g., binder clogged). This information goes into the AI's decision process for what to do next (e.g., if a print failed, it might reschedule that experiment on a different printer or adjust parameters to fix the issue).

Phase 2: Command and Control Automation

Goal: Allow the AI platform to **directly command equipment**, including starting prints, controlling robotic elements, and orchestrating the entire lab sequence without manual intervention, while using Desktop Metal systems within safe operating bounds.

- **Direct Print Control via APIs/SDKs:** In this phase, we move beyond just submitting jobs through the GUI. We collaborate with Desktop Metal (if possible) to gain access to a lower-level **printer API or SDK**. Many industrial machines have an interface for integration (for example, a REST API to start/stop a job, or a socket connection to feed G-code or job files). Using such an interface, the AI can schedule prints on each machine directly. It could decide which printer is optimal for a job (based on availability or required material) and dispatch the print to that specific machine. It can also set key parameters on the fly (e.g., binder saturation level, layer thickness if variable, etc.). Real-time control (layer-by-layer intervention) is still limited, but between print runs the AI has full control.

- **Robotics Integration (via ROS):** By Phase 2, the robotic arm and other handling systems are operational. Integration strategy involves deploying a **ROS (Robot Operating System) network** that includes the AI's agent node and the robot's control node. The AI issues high-level motion requests, and the ROS stack handles execution and returns success/failure. For example, after a print completes, the AI sends a command like "robot, fetch build box from Printer 2." A pre-written ROS action server knows how to do this (with known coordinates for Printer 2's build area, etc.) and will execute the pick-up. The AI will have logic to verify the action (checking a sensor that the part was picked, etc.). If something goes wrong (robot reports an inability to grasp, etc.), the AI can decide to retry or alert a human. The integration ensures **synchronization**: printers are polled to ensure they are cool and safe to touch before the robot reaches in, etc., to avoid hazards.

- **Unified Workflow Orchestration:** In this stage, the **control software (AI agent) becomes the primary orchestrator** of the entire workflow. It uses a *workflow engine* that knows the sequence: Print → Depowder → Sinter → Test. This engine coordinates the timing: for instance, right after printing, it can immediately send the part to depowdering if the furnace is ready, or if the furnace is busy, the part might wait (or the AI might print another part in the meantime). The integration here involves **schedule optimization** algorithms to maximize throughput while avoiding conflicts (two robots trying to use the same station, etc.). The AI can treat the lab like a CPU scheduling tasks, using strategies to pipeline processes (e.g., start sintering one batch while another batch is printing).

- **Data Federation and IoT:** At this point, all equipment (printers, furnace, environmental sensors, robots) are connected to a common **industrial IoT platform** (which could be part of the Live Suite or a custom solution). We ensure that everything speaks a common protocol (for example, using MQTT topics for events: "Printer1/JobComplete" or "Furnace/TempReached"). The AI subscribes to these events rather than actively polling, making the system more scalable and real-time responsive. This decoupling via events also makes it easier to integrate new machines – one just needs to start publishing its state in the agreed format.

- **Safety and Sandbox Constraints:** Integration strategy also involves setting constraints to ensure that the AI's control remains within safe limits. For example, we implement a **virtual safety controller** that will intercept any command that might harm the machine or is out of bounds (like trying to start a print without powder loaded, or commanding the furnace to a temperature above its rating). Initially, these rules can be hard-coded or provided by the machine vendors. Over time, the AI can learn these as well (e.g., from a manufacturer’s documentation encoded into its knowledge base), but we keep a fail-safe layer to prevent accidents. We also maintain **manual override** capabilities – at any time an operator can hit an e-stop or software stop that the AI will recognize and immediately pause actions.

Phase 3: Advanced Integration and Optimization

Goal: Achieve **full autonomy with optimization** – the AI not only runs the lab equipment but actively **optimizes their performance** and extends integration to new capabilities (like multi-material experiments, cloud integration, etc.).

- **Dynamic Parameter Optimization:** With deeper integration, the AI can vary printer parameters *within* a single print or adapt between prints in a fine-grained way. For instance, the AI could perform a **print parameter sweep**: in one print job, different sections of the build plate get different settings (binder amount, layer height) to see which yields the best density after sintering. To do this, the integration must allow changing settings per part – which might involve generating multiple build files or using printer scripting if supported. We will work with Desktop Metal’s software to see if such customization is possible (for example, some binder jet software allows setting different “recipes” for different parts in the same job). If not natively supported, the AI can break the experiment into multiple sequential prints, adjusting settings each run. Over multiple experiments, a closed-loop parameter optimization is achieved (the platform essentially conducts something like a response surface optimization on the print parameters for a given material).

- **Robotic Calibration and Learning Integration:** As the robot and other hardware operate repeatedly, the integration includes a calibration routine. The AI will periodically run checks (using camera to verify known positions, measuring if a part is exactly where expected after an operation, etc.) and auto-calibrate offsets. If any drift is found (say a printer’s coordinate system shifted slightly after maintenance), the AI can apply software compensation. This is integrated by having a **calibration service** that the AI can call, which uses sensors to adjust the robotic coordinate frames.

- **Integration of Additional Lab Equipment:** In advanced stages, the platform might integrate *additional processes* like mixing of powders (for creating new alloy compositions from base powders), or heat treatments, or even entirely different printing technologies (maybe a polymer printer for making a mold, etc.). The strategy for integration remains the same: abstract the device behind an API and event model, then teach the AI how to use it. For example, an automated powder mixer could be integrated where the AI specifies “50% Fe powder, 50% Co powder” and the mixer dispenses and blends those, then the blend is fed into the printer. This would likely require custom hardware integration, but since our architecture is modular, adding a “Mixing station” node is straightforward. The AI just gains a new action it can invoke (“mix powders”), and the workflow engine adds a step in the sequence when a new composition powder is needed.

- **Cloud and Scalable Integration:** In the final vision, we foresee connecting multiple labs or using cloud resources. The integration strategy includes a **cloud-based supervisory system** where multiple autonomous labs (perhaps at different locations or units) report to a central system. This central brain aggregates results and might redistribute tasks. For example, if one lab finds a promising material, another lab could independently verify it. Or certain computations (like heavy simulations) are sent to cloud HPC and results integrated back. While this is more of an expansion than integration with Desktop Metal specifically, it highlights that the system is designed to **scale out**. Desktop Metal printers are already designed with Industry 4.0 in mind, meaning multiple printers can be managed in a fleet via Live Suite desktopmetal.com. We tap into that capability to manage the printers collectively: the AI sees a pool of manufacturing resources rather than one machine at a time.

- **Vendor Collaboration:** Throughout integration, close collaboration with Desktop Metal's engineering is assumed to get the most out of their machines. By Phase 3, we might work on **custom firmware tweaks** or special modes for research (e.g., an "open material mode" where the printer accepts non-certified materials and allows custom parameters – something normally locked down in commercial use). We ensure any such collaboration follows machine warranties and safety guidelines, effectively turning these printers into part of a specialized self-driving lab apparatus. The benefit for the vendor is demonstrating how their systems can be used in cutting-edge automated research.

In summary, the integration strategy begins by treating the Desktop Metal printers as **smart black boxes to observe**, then gradually turns them into **controllable instruments** in the autonomous lab. By establishing data links first, we ensure the AI is well-informed. Then by adding control, we enable automation. Finally, by optimizing and expanding capabilities, we push toward true autonomy and efficiency. Each phase builds on the previous, reducing risk – for example, by the time we allow the AI to fully control a printer, we have already seen many prints and understand failure modes, so the AI's decisions are grounded in experience. This phased integration ensures a smooth transition from today's manual or semi-automated lab operation to the envisioned *lights-out*, AI-run materials discovery lab of the future.

Development Roadmap

Building the world's first fully automated, AI-driven materials discovery platform is an ambitious undertaking. We outline a development roadmap with clear milestones to progressively achieve the vision. The roadmap is divided into stages, each adding capabilities and complexity in both software and hardware. This phased approach ensures validated learning at each step and de-risks the path to full automation.

Stage 0: Foundational Setup (Months 0-6)
Focus: Set up basic infrastructure and achieve digital connectivity.

- **Lab Setup & Instrumentation:** Install the Desktop Metal printers (Shop, P1, X25 Pro, X160 Pro) and ancillary equipment in the lab. Set up the sintering furnace and a basic depowdering station. Equip the printers with additional sensors if needed (e.g., a camera in each printer if not already provided). Ensure all devices are on the network.

- **Data Pipeline Initialization:** Implement the initial data logging system. For each printer, establish a way to retrieve print job data (via API or even log file scraping if necessary). Test that when a print is run via standard procedure, data appears in the central database. Also, integrate simple sensor logging (e.g., a standalone temperature/humidity sensor in the lab environment, just to validate IoT connectivity).

- **LLM Agent MVP (Minimal Viable Product):** Deploy a basic version of the AI Orchestrator. At this stage, it can be a simplified loop that *suggests experiments* but perhaps doesn't execute them. For example, using historical or simulated data, the LLM can generate a materials suggestion and analyze a hypothetical result. This is mostly to test the prompt design, self-reflection mechanism, and integration of any surrogate models. We might use published case studies (like known improvements of a material) as a sandbox to see if the LLM can "discover" them in theory. No real prints yet – this is a dry-run of the AI logic.

- **Initial Integration Testing:** Have the AI agent send a dummy print command to a test server to simulate job submission. This ensures our pipeline from AI decision → command generation → reception by hardware interface works. We might, for instance, intercept the command and manually start a print to compare if the AI's intended parameters match what the printer received.

Stage 1: Automated Experimentation Loop (Months 6-18)
Focus: Achieve a functioning closed-loop with one printer and limited human intervention.

- **Print & Sinter Loop Automation:** Enable the AI to go from idea to printed sample to measured result in one loop on a single machine (say the Shop System or P1). This includes the robot (or a human proxy initially) transferring the part to furnace and running a sintering cycle. For early development, we might still have a human unload the furnace and measure properties, but the AI will consider that as part of the loop (with the human feeding back the measurement into the system). Essentially, by the end of Stage 1, if we give the platform a target (like maximize density of a given alloy by varying sintering temperature), it should autonomously conduct a series of prints and sintering runs to converge to an answer.

- **Robotic Arm Integration (Prototype):** Introduce the robotic arm in a limited capacity. Possibly start with a single task like moving a build plate from printer to a staging area.

Gradually test more complex sequences. At this stage, the robot might be tele-operated or semi-automated – e.g., the AI tells the robot “go” and a low-level script executes a fixed motion. The aim is to debug any mechanical issues and ensure safe operation before it’s fully autonomous.

- **Basic AI Decision-Making:** The LLM agent is now making real decisions. Implement a basic version of the self-reflection: after each experiment, log the AI’s reasoning about what happened. Begin training/updating surrogate models with real data from prints (for instance, train a simple linear regression or neural net on any property results we get). By the end of this stage, the AI should demonstrate adaptive behavior – e.g., if two experiments in a row show that “adding X reduces performance,” the third experiment should try something different, indicating it learned from those outcomes.

- **Use Case Dry Run:** Attempt a simplified version of a target use case with the system. For example, try to optimize a known property like density or hardness for a well-characterized material (like 17-4 PH stainless steel, which Desktop Metal uses often). This is to validate the loop. We wouldn’t necessarily discover something new here (since the material is known), but we verify the platform can reach a known optimum on its own. It’s a form of calibration for the AI in the physical world.

Stage 2: Multi-Machine Coordination & Expanded Autonomy (Months 18-30)
Focus: Scale the system to multiple printers and remove most human inputs.

- **Fleet Coordination:** Enable the AI to manage multiple printers concurrently. By now, integration with all target Desktop Metal printers is done. The AI can decide, for example, to use the P1 for rapid screening of 5 variants, then use the X25 Pro to print a larger sample of the best candidate for thorough testing. Implement scheduling logic to avoid conflicts (two jobs needing the furnace at once, etc.). The platform should handle queuing – perhaps the Shop System runs an experiment overnight while the P1 does quick ones during the day.

- **Full Robotic Lab Automation:** Finalize robotic integration so that **no human handling is required**. The robotic arm (or arms) now cover the full workflow: removing prints, depowdering (maybe with a mechanized depowder unit by now), loading the furnace, unloading it, and even delivering parts to testing devices. Achieve “lights-out” operation for at least a continuous 24-48 hour trial, where the AI runs a series of experiments without intervention llnl.gov. This will likely surface issues (like a powder spill or a sensor fault) which we use to further harden the system.

- **Advanced AI Enhancements:** Integrate more sophisticated reasoning tools. For example, introduce a **retrieval-augmented generation** step where the AI can query a literature database for guidance (important for tackling harder use cases). Incorporate domain-specific LLM prompt tuning (perhaps a fine-tuned LLM on materials science text) to improve accuracy on technical queries. Also, emphasize **failure handling** in AI: teach it fallback strategies (if a printer goes down, switch tasks to another; if a measurement seems off, repeat or recalibrate instrument).

- **Model-Driven Experimentation:** At this stage, the surrogate models and data-driven suggestions should take a bigger role. The AI might run virtual screening in parallel to physical experiments. We may integrate a Bayesian optimization loop around the AI's suggestions to ensure systematic exploration vs exploitation. The outcome: the experiment selection becomes more optimal, focusing on the most informative experiments.

- **Safety & Ethics Review:** Before fully unleashing the platform on novel R&D problems, conduct a thorough review of safety logs and decision-making. Ensure the AI's autonomous choices stay within ethical and safe bounds (for example, if asked to use a toxic element, does it have protocols? If trying an experiment likely to fail dangerously, does it catch that?). This might involve adding more constraints or an oversight module.

Stage 3: Generalization & Targeted Discoveries (Months 30-48)
Focus: Tackle the complex use cases and demonstrate generality.

- **Use Case 1 – HRE-free Permanent Magnet Discovery:** Deploy the platform on the heavy rare-earth-free magnet problem. By this time, the system should be fully equipped: multiple alloy compositions can be printed via perhaps mixing powders (if a powder mixing system is integrated by now) or using pre-alloyed powders. The AI will explore compositions and processing (like different sintering profiles, or maybe post-sinter heat treatments) to find a magnet that meets criteria (high coercivity without Dy or Tb doping). Expectation is to iterate dozens or hundreds of experiments rapidly. A successful milestone would be identifying an alloy that matches or exceeds a baseline magnet performance without heavy rare earths. Document the discovery process – this will serve as a high-profile validation of the platform's capabilities.

- **Use Case 2 – Advanced Ferromagnets or Soft Magnet Alloys:** In parallel or after, run a campaign on ferromagnetic alloys (e.g., finding a high-saturation magnetic material for electric motors that uses less cobalt). This showcases the platform's ability to optimize a different kind of magnetic property. It might involve different testing apparatus (measuring magnetic saturation and core losses), which by now would be integrated. Achieve a result such as identifying an alloy or process (maybe a specific cooling rate, etc.) that yields improved magnetization. This highlights generality in the magnetism domain.

- **Use Case 3 – Solid-State Sodium-Ion Electrolytes:** Simultaneously or subsequently, challenge the platform with a very different domain (electrochemistry vs metallurgy). Set up the system to explore solid electrolyte materials (likely ceramics or glassy sulfides). This may involve integrating a ball-milling or solution synthesis step if purely printing is not sufficient (alternatively, possibly using the printers to create pellet samples from blended powders). The AI’s task would be to discover a composition or fabrication process that yields high ionic conductivity and stability. Success could be measured by finding an electrolyte with conductivity approaching that of liquid electrolytes or significantly improved stability. Since this is a different field, it will prove that the platform is not narrowly tailored but **adaptable to broad materials R&D challenges**.

- **Scalability and Throughput Improvements:** During this stage, we also aim to speed up the cycle. Identify bottlenecks – perhaps sintering is the slowest step. We could introduce multiple small furnaces to run in parallel or use faster heating methods for small samples. The AI could then manage parallel experiments truly concurrently (e.g., four experiments at once in four furnaces). By month ~42, we want the lab to be running at full capacity, executing experiments round the clock. A metric could be “experiments per week” – we target an order-of-magnitude increase over a traditional lab. For instance, if a human lab might do 5-10 iterations in a week, the autonomous lab could do 50-100+ in the same time.

- **Refinement of AI models:** With a trove of data now, revisit the AI models for improvement. Possibly retrain the core LLM agent with domain-specific data or fine-tune its self-reflection prompts to correct any observed flaws. Also refine surrogate models to be highly predictive within the domain the platform has explored – effectively creating a *digital twin* of the experiment space that the AI can use even more heavily to guide decisions.

Stage 4: Fully Autonomous Self-Driving Laboratory (Months 48+ and beyond)
Focus: Long-term evolution and self-maintenance.

- **Autonomous Maintenance:** Develop routines for the system to check itself and perform basic maintenance or requests for maintenance. For example, the AI detects if a printer needs a new binder cartridge and can alert a human or even trigger an automated refill if that infrastructure exists. Similarly, the robot might do a daily calibration check or a routine cleaning of powder from the floor if a spill is detected (assuming a cleaning robot can be integrated).

- **Continuous Learning and AI Evolution:** Implement an *online learning* approach where the AI’s models continue to improve with each new batch of data. Also, the AI can be updated/upgraded (for instance, if GPT-5 becomes available, integrate it and validate that it improves performance). The architecture is made flexible to accommodate such swaps.

- **Remote Collaboration and Scaling:** Allow remote researchers to utilize the platform (with proper controls). The platform could be accessible via a cloud interface where a researcher anywhere can submit a materials discovery query, and the AI schedules it into the lab’s queue. Over time, multiple such labs could network to share data – the roadmap in a visionary sense sees **distributed self-driving labs collaborating**, orchestrated by LLM agents that communicate insights to each other. This would truly accelerate materials R&D globally.

- **Documentation of Discoveries:** Finally, the platform itself could auto-document its findings. By Stage 4, the AI might be trusted to draft scientific reports or patent applications based on its discoveries (with human oversight). This showcases not only discovery but also the ability of AI to communicate new knowledge – completing the loop from hypothesis to documented result. For example, if the platform discovers a new Na-ion electrolyte, it could write up the synthesis process and performance in a report, citing the data it gathered – essentially acting as a robotic scientist that publishes its work.

This roadmap, spanning roughly 4+ years, is aggressive but attainable with focused development. By breaking the challenge into stages, we ensure at each milestone we have a working subset of the vision. Early successes (even partial automation) will build momentum and justify further integration. Ultimately, by following this roadmap, we expect to achieve the end goal: **a fully autonomous AI-driven materials lab that can operate 24/7, greatly reducing the time and cost to discover and optimize new materials** llnl.gov. The implications include not just solving the specific use cases outlined, but fundamentally changing how materials research is conducted, heralding an era of self-driving laboratories for a wide array of scientific discovery tasks.

Research Use Cases and Scenarios

To illustrate the platform's capabilities and generality, we consider several representative materials R&D use cases. These scenarios demonstrate how the system would approach different challenges – from metallurgy to energy storage – highlighting the integration of AI planning, autonomous experimentation, and the iterative improvement loop. Each use case was chosen because of its high scientific and industrial importance, and the difficulties involved make them ideal candidates for an autonomous discovery approach.

Use Case 1: Heavy Rare-Earth-Free Permanent Magnets

Challenge: Permanent magnets like Neodymium-Iron-Boron (Nd-Fe-B) are critical in motors and generators, but they typically require heavy rare earth (HRE) elements such as Dysprosium (Dy) or Terbium to maintain performance at high temperatures. These HREs are expensive and subject to supply risk ameslab.gov. The goal is to discover or optimize magnet alloys that **eliminate heavy rare earths** while retaining high magnetic performance (high coercivity, high remanence, and high Curie temperature).

Approach: The autonomous platform tackles this by exploring alternative alloy compositions and processing routes:

- **Knowledge Initialization:** The AI starts with literature knowledge of known magnet systems. For example, it knows that $Nd_2Fe_{14}B$ is the strongest base magnet phase, and that adding Dy increases coercivity by improving anisotropy field but can be removed if grain boundaries are engineered properly ameslab.govameslab.gov. It also knows of candidate HRE-free systems (e.g., cerium-based magnets, or entirely different phases like Mn-Bi, Alnico, or Fe-Ni alloys with special treatments). This knowledge is part of its prompt/context.

- **Experimental Plan:** The AI might propose initially to experiment with substitutions in the Nd-Fe-B system that could enhance high-temperature coercivity. For instance, it may try adding alloying elements like cobalt (to increase Curie temperature) or certain metalloid elements to pin domain walls. Simultaneously, it might explore completely rare-earth-free systems like Mn-Bi or Fe-Co alloys known for good magnetism but needing optimization ameslab.gov.

- **Printing & Processing:** Using the binder jet printers, the platform can create small samples of these alloys. If powder of the exact composition isn't available, the AI can blend elemental or master alloy powders in the desired ratios (the system could print a "pseudo-alloy" by co-depositing mixed powders that will then alloy during sintering). For instance, to test Nd-Fe-B with 2% substituting an element X, it mixes those powders and prints a pellet or other simple geometry.

- **Post-Processing:** Magnet samples need specific post-processing: sintering to near full density, and in some cases, heat treatments like grain boundary diffusion or annealing. The AI will use the furnace to sinter the printed magnets at, say, 1050°C in vacuum. It could also integrate a step to **magnetically align** grains if needed (though for isotropic tests it may skip this).

- **Testing:** After production, each sample's magnetic properties are measured. The platform likely has a **magnetometer** setup or a loop tracer to get the hysteresis curve. Key metrics like coercivity (resistance to demagnetization) and remanent magnetization are recorded at both room temperature and elevated temperature (to ensure performance at e.g. 150°C for motors ameslab.gov).

- **Iterative Optimization:** Suppose the AI tries Mn-Bi as suggested by literature (since Ames Lab found MnBi can have coercivity that increases with temperature ameslab.gov). Early tests might show moderate magnet performance but lower magnetization than NdFeB. The AI reflects on this: it notes Mn-Bi's positive temperature coefficient for coercivity and its lack of rare earths is promising, but magnetization is lower and the material is brittle. It then hypothesizes ways to improve magnetization (maybe by adding iron or cobalt to Mn-Bi) or to improve microstructure (perhaps adding a third element to refine grain boundaries). It runs those experiments.

- **AI Reasoning:** Throughout, the AI uses its **self-reflection** to guide the search. If a certain path (like Co substitution in NdFeB) isn't yielding improvements, it will analyze the reasons – e.g., "Co raised Curie temperature but also reduced anisotropy, net effect is insufficient coercivity." It may then pivot strategy, perhaps focusing on microstructure control: e.g., "Instead of chemistry, try a different sintering profile to create smaller grains," since smaller grains can improve coercivity by preventing magnetization reversal cascades.

- **Outcome:** Ideally, the platform converges on one or more candidates. For example, it might discover an alloy Nd-Fe-B with some amount of another element (or a completely new compound like a specific Fe-Co-B alloy with nanoscale additives) that achieves high coercivity without Dy. Or it might validate that a combination of processing steps (like a specific grain boundary diffusion of a non-heavy element) gives the needed performance. It will have supporting data – micrographs showing isolated grains (like how Ames Lab isolated MnBi grains with polymer coating in their research ameslab.gov), and magnetic measurements to prove the achievement.

- **Generalization:** The knowledge gained (like "element X can substitute for Dy's role if grain size is under Y micrometers") is fed back into the knowledge base, so the AI can apply it to future magnet development projects as well.

Through this use case, the platform demonstrates its ability to solve a **multivariate problem (composition + process)** in a critical materials domain. The elimination of heavy rare earths from magnets has huge economic and strategic implications eta-gp.com, and doing so requires navigating a complex design space – exactly what the autonomous platform is built for.

Use Case 2: Advanced Ferromagnetic Alloys (Soft Magnets for Electric Motors)

Challenge: Soft magnetic materials (like silicon steel or amorphous alloys) are used in transformer cores and motor laminations. There is ongoing demand for materials that have higher saturation magnetization and lower core losses for improved motor efficiency, potentially at higher operating temperatures. Traditional high-performance soft magnets often contain significant cobalt (e.g., Fe-Co-V alloys) which is costly, or they face trade-offs like brittleness or difficulty in manufacturing. The goal here is to discover or optimize a **ferromagnetic alloy with high saturation magnetization and low hysteresis loss**, while using earth-abundant elements.

Approach: This use case tests the platform's capability in a different regime of magnetism – focusing on saturation magnetization (a property linked to composition and structure) and minimizing energy loss (related to electrical resistivity and domain wall motion).

- **Defining Objectives:** The AI is instructed to maximize saturation magnetization (Ms) and minimize coercivity (since for soft magnets we want low coercivity) and core loss. It may also incorporate resistivity as a factor (higher resistivity reduces eddy current loss in AC applications). These are multi-objective goals, so the AI will likely aim for a good balance (for example, an alloy with very high Ms but too high coercivity might be ruled out).

- **Experimental Space:** The AI explores iron-based alloys with various additives:
  - It knows pure iron has very high Ms but is soft and has high eddy losses. Adding Si increases resistivity (lowering losses) but too much Si makes it brittle.
  - It considers Fe-Co alloys: cobalt increases Ms (Fe-Co has one of the highest known saturation values) but Co is expensive. Perhaps the AI tries to reduce Co content by adding small amounts of other elements to maintain performance.
  - It might also consider newer alloys like Fe-Ni (permalloys) or iron-based amorphous/nanocrystalline alloys that achieve low losses.
- **Fabrication Strategy:** The platform can 3D print simple toroidal or rectangular ring samples that can be used for magnetic testing (a toroid is useful for measuring core loss under AC excitation). Using binder jet, printing iron alloys is feasible (and the sintering step can densify them). If certain compositions are not directly printable due to brittleness (e.g., high Si steel might crack when sintered), the AI might note processing adjustments like using a transient liquid phase or adding a sintering aid.
- **Testing:** For soft magnets, testing involves magnetization curves and loss measurement. The platform might integrate a setup with a drive coil and sense coil around the toroidal sample to measure hysteresis loops at various frequencies. The AI can derive properties like saturation magnetization, permeability, and core loss from these loops.
- **Iterative Discovery:** Suppose the AI starts with an Fe-50%Co alloy for highest Ms. It finds great Ms but also observes that coercivity is a bit high and the material is expensive. It then tries adding a small percent of phosphorous or boron to induce a nanocrystalline structure upon heat treatment (a known technique in soft magnetic alloys). This could lower coercivity by creating smaller magnetic domains. The prints are done, and a specialized heat treatment (maybe a rapid anneal) is applied by the furnace.
- **Outcome Analysis:** The AI might discover that an Fe-Co-B-P alloy achieves nearly the high Ms of FeCo but with much lower coercivity after proper annealing (essentially creating a two-phase nanostructure). Or it might find that Fe-Si with a little bit of aluminum yields a good trade-off of performance vs ductility, enabling higher Si content than traditionally used but still processable.

- **AI Reasoning & Transfer:** During this, the AI uses multi-objective reasoning. It might employ a Pareto front approach – identifying sets of experiments that are optimal trade-offs. If one sample has very low loss but also lower Ms, and another has very high Ms but higher loss, it will note both and possibly suggest further experiments that try to get the best of both (for instance, combining aspects of both approaches). The agent could even be prompted to "improve Ms without increasing loss" explicitly, guiding it to creative solutions like microstructural engineering.

- **General Outcome:** The platform would ideally produce one or two candidate compositions and process recipes that industry can further develop. E.g., it might propose "Fe-8%Si-4%Al alloy, processed by fast sintering and controlled cooling, yields high saturation and 20% lower losses than conventional electrical steel." That result could be directly useful for motor manufacturing. More importantly, it shows how the platform can optimize **electro-magnetic functional properties** which involve subtle interplay of composition and microstructure.

This use case emphasizes the platform's strength in **optimizing materials for multiple properties** and handling continuous variables (like percentages of alloying elements, processing times) in its search. It also stresses how the AI can manage complex trade-offs, a scenario where there isn't a single objective but a balance – perfect for its ability to reason and weigh options.

Use Case 3: Solid-State Sodium-Ion Battery Electrolytes

Challenge: Solid-state sodium-ion batteries (SSSBs) require solid electrolytes that conduct Na^+ ions efficiently at room temperature, are stable (both chemically and mechanically), and easy to fabricate. The current options (such as NASICON-type oxides or sulfide glasses) often suffer from either low ionic conductivity, instability with electrodes, or difficult processing (brittleness, requiring high pressure, etc.) sciencedaily.com. The goal is to discover or optimize a **solid electrolyte material for Na-ion batteries** that has high ionic conductivity (ideally >1 mS/cm), stability against Na metal, and can be made in dense, thin layers.

Approach: This is a chemistry-centric problem, different from metallic alloys. The platform will explore **ceramic and glassy materials** compositions and their processing:

- **Compositional Space:** The AI considers known families: e.g., *NASICON-type* (sodium superionic conductors like $Na_{1+x}Zr_2Si_xP_{3-x}O_{12}$), *sulfide glasses/crystals* (like Na_3PS_4, Na_3PSe_4, etc.), and *halide or borohydride electrolytes*. It also has the freedom to consider novel mixtures or doping of these. The chemical search space is huge – but the AI can narrow it by logic (it knows ionic radius and polarizability matter, etc.) and perhaps by using prior data (maybe from computational screening studies nature.com).

- **Fabrication via Printing:** One way to make solid electrolytes is to cold-press powders or sinter them into pellets. Our platform can leverage the printers in a creative way: for example, use the binder jet printer to shape a pellet or even a complex structure (like a thin-walled container shape) from electrolyte powder, then sinter it to form a dense part. Alternatively, if materials are sensitive (sulfides are air-sensitive), we might print within a controlled atmosphere or simply prepare powders and load them into a furnace with pressing dies. We might also use the polymer printers (if any in the lab) to print a mold or scaffold that is later infiltrated with an electrolyte solution and then converted to a solid.

- **Experiment Plan:** The AI might start with a known decent conductor like Na_3PS_4 (a sulfide). It prints and sinters a sample, measures ionic conductivity by placing it in a test cell (blocked electrodes method or similar). Suppose it gets 0.5 mS/cm at room temp. The target is >1 mS/cm, and improved stability. It then tries modifications: e.g., partial substitution of sulfur with selenium ($Na_3P(S/Se)_4$) to see if a larger lattice might boost conductivity (selenium often increases ionic conductivity at cost of lower glass transition – the AI would reason about such trade-offs).

- **Testing:** The platform uses an electrochemical impedance spectroscopy (EIS) setup. It attaches electrodes (maybe sodium metal or non-blocking electrodes) to the sample and sweeps frequency to get an impedance spectrum. The AI extracts the bulk ionic resistance and calculates conductivity. It also performs stability tests: for instance, hold a sodium metal electrode against it and see if any reaction (increase in interfacial resistance) occurs over time. If we have a solid-state cell assembly capability, it could even assemble a simple cell and cycle it a few times to test endurance.

- **Iterative Search:** The AI will apply both domain knowledge and experimental results. For example, it might recall that introducing certain glass formers like Boron could help form a glassy superionic conductor. It might propose exploring compositions like adding $Na_2B_{12}H_{12}$ or other borates into the sulfide to enhance flexibility (some literature suggests complex anions can improve Na conductivity). The system tries mixing these additives and sintering. Or it might explore completely different systems after some tries – e.g., NASICON oxides which require higher sintering temperature but maybe the AI figures out a way to lower sintering temperature by adding a sintering aid (like a small amount of glass).

- **Use of Surrogate Models:** Given the large compositional space, the AI will likely lean on any available computational model. Perhaps it uses a machine-learned predictor for ionic conductivity that was trained on known Li/Na conductors to guide which compositions to test first. Or it uses first-principles on-the-fly: e.g., it might call a DFT calculation to compute migration barriers in a hypothetical crystal it comes up with, to screen viability before experimental attempt.

- **Outcome:** The platform could discover, for example, a modified sulfide electrolyte that after sintering has a conductivity of ~2 mS/cm (which is very high for Na, perhaps achieved by a specific mix of Sulfur and Selenium and some glassy fraction to aid grain boundaries). Or it might find an entirely new composition, like a certain Na-rich halide that was relatively unexplored, that shows good stability and decent conductivity when processed a certain way. It will also determine **processing conditions** that matter – perhaps it finds that sintering in an Ar/H_2 atmosphere yields better conductivity (maybe due to reducing some impurities).

- **Generalization and Use:** Once a good candidate is found, the AI can also attempt to optimize how to manufacture it as a thin layer (since practical batteries need thin electrolytes). It might, for instance, use the printers to create a thin lattice that sinters into a dense membrane, or figure out a method to tape-cast and sinter using the equipment. This goes slightly beyond discovery into process engineering, which the platform can also handle through experimentation.

This use case highlights the platform's ability to work with **non-metallic, chemically complex materials** and to innovate in chemical composition as well as processing. Solid electrolytes are a hot research area, and the platform's autonomous approach could rapidly sift through many combinations (something that manually would be extremely time-consuming). Given that researchers have identified some high-conductivity compositions recently sciencedaily.com, the AI could take those as starting points and then **fine-tune and stress-test them** (for example, improving stability or making them easier to produce). The result would be a deeper understanding and possibly a patentable new material for sodium batteries discovered in a fraction of the time of traditional trial-and-error.

Conclusion of Use Cases: These scenarios demonstrate the versatility of the proposed platform. It is equally adept at metallic alloy optimization (permanent magnets, ferromagnets) as it is at ionic compound discovery (battery electrolytes). In each case, the platform follows the same fundamental cycle – propose, make, measure, learn – but tailors the specifics (tools, criteria, domain knowledge) to the problem at hand. By incorporating large language models and a rich set of tools, the system can ingest decades of human knowledge in these fields and operate with creativity and diligence, far faster than a human team could.

Each successful use case will not only yield valuable results (better magnets, better batteries) but also serve as a proof-of-concept to persuade stakeholders of the platform's transformative potential. This approach could be extended to countless other materials challenges (catalysts, pharmaceuticals, polymers, superconductors, etc.), positioning the platform as a **generalizable solution for autonomous R&D**. The inclusion of advanced AI techniques (LLM reasoning, multi-modal agents) and full integration with manufacturing hardware truly defines a new state-of-the-art: a lab that *thinks*, *acts*, and *learns* to accelerate innovation.

By ensuring clarity, specificity, and thorough documentation at each stage (as done in this specification), R&D and engineering teams can align on the vision and systematically build this groundbreaking platform – making the once-futuristic dream of a self-driving lab for materials discovery a reality in the very near future llnl.govllnl.gov.

Sources:

1. Jia, S., Zhang, C., & Fung, V. *LLMatDesign: Autonomous Materials Discovery with Large Language Models*. arXiv preprint 2406.13163 (2024). themoonlight.iothemoonlight.io

2. LLNL News (July 21, 2025). *Self-driving lab to automate the discovery of novel alloys* – Describes the APEX platform combining robotics and ML for alloy design llnl.govllnl.gov.

3. Huang, W. et al. *VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models*. CoRL 2023. (Demonstrates using GPT-4 to plan robot actions) openreview.net.

4. Emoekabu, B. et al. *ChemBench: Evaluating the chemical knowledge of LLMs*. Nature Chemistry 17, 1027 (2025). (Finds LLMs can outperform human chemists on average, but with some gaps) nature.com.

5. Nandi, S. (Apr 4, 2025). *What Happens When You Put ChatGPT-4 Inside a Robot?* AZoRobotics News. (Discusses ELLMER, an embodied LLM robot with vision and feedback) azorobotics.comazorobotics.com.

6. Desktop Metal, Inc. *Live Suite™ – End-to-End Software Hub for Additive Manufacturing*. (Product documentation) desktopmetal.comdesktopmetal.com.

7. Ames Laboratory News (Apr 30, 2025). *Scientists develop rare earth free magnet for use in industrial motors*. (Highlights need for HRE-free magnets and a Mn-Bi solution) ameslab.govameslab.gov.

8. Osaka Metropolitan Univ. via ScienceDaily (Apr 11, 2024). *Discovery brings all-solid-state sodium batteries closer to practical use*. (Reports a new Na superionic conductor) sciencedaily.comsciencedaily.com.

Technology Specification for ARC's AI-driven materials discovery and autonomous manufacturing platform, revised to incorporate best practices from current Agentic Design Patterns literature and adjacent work on scientific LLM systems and robotics.

```None
Scope: End-to-end platform for data-driven materials discovery
(near-term: data transfer + feedback loops; long-term: full
robotic handling, print parameter optimization, sintering,
depowdering) seamlessly integrated with Desktop Metal Shop
System™, Production System™ P-1, X25 Pro™, and X160 Pro™.
```

1) Executive Overview

ARC will build a **graph-orchestrated, multi-agent system** that uses *planning, tool-use, reflection, and multi-agent collaboration* to generate hypotheses, design experiments, run print/sinter cycles, measure outcomes, and continuously improve material/process candidates. The system emphasizes:

- **Agentic patterns** (Planner-Executor-Critic, Self/Cross/Human Reflection, Tool/Agent Registry, Single- vs Multi-path planning, One-shot vs Incremental querying, Guardrails). ar5iv

- **Structured outputs & typed tool calls** to reliably drive instruments and workflows. TECHCOMMUNITY.MICROSOFT.COM

- **Tight integration with Desktop Metal Live Suite & Live Sinter** for build prep, simulation, telemetry, and fleet control. Desktop Metal+1

- **Closed-loop discovery** proven in recent LLM-for-materials frameworks (iterative modification → validation → self-reflection → next step).
 LLMatDesign: Autonomous Materia…

- **Operational safety & reliability**: typed contracts, guardrails, watchdogs, incident recovery, and human-in-the-loop (HITL) where warranted by risk and uncertainty in LLM reasoning. ar5iv

---

2) Design Tenets (from Agentic Patterns)

1. **Planner–Executor–Critic (PEC) loop**
   Decompose objectives → generate structured plans → execute via tools → critique & revise. Implements *Planning*, *Tool-Use*, and *Reflection* patterns. Stanford University

2. **Self, Cross, and Human Reflection**

   - *Self-reflection* (the agent critiques its own step)

   - *Cross-reflection* (a peer/alternative model critiques)

   - *Human reflection* (HITL approval gates for higher-risk actions).
     These are codified design patterns for improving reasoning certainty and explainability. ar5iv

3. **Single-path vs Multi-path Planning**
   Use *multi-path* (branching hypotheses) when exploration is desired; *single-path* for exploitation/production. ar5iv

4. **One-shot vs Incremental Model Querying**

   - *One-shot* for low-risk, low-variance tasks (cost-efficient).

   - *Incremental* for long-horizon workflows that benefit from updated context and verifiability. ar5iv

5. **Tool/Agent Registry**
   Typed registry advertising capabilities, schemas, states, cost/time SLAs, and safety class for every tool/agent (printers, furnaces, metrology, simulators, databases). ar5iv

6. **Multimodal Guardrails & Approval Gates**
   Validate inputs/outputs, enforce constraints, and require HITL for actions with elevated process risk (e.g., new sinter schedules or powder/binder combinations). ar5iv

7. **Structured Output Everywhere**
   All agent communications and tool calls use typed JSON schemas (Pydantic-style), improving orchestration and error handling. TECHCOMMUNITY.MICROSOFT.COM

Note: Public coverage of the "Agentic Design Patterns" work indicates a comprehensive set of ~21 patterns and emphasizes pattern composition (planning + tool use + multi-agent + reflection + safety + memory). PPC Land

---

3) System Architecture

3.1 Logical View

- **Goal Ingestion & Governance**
  - Research briefs & objectives (e.g., "HRE-free permanent magnet with BH_max ≥ target", "Na-ion solid electrolyte σ ≥ target at 25 °C").
  - Compliance policy & risk profile.
- **Agent Orchestrator (Graph Runtime)**
  - State machine / DAG execution with recoverable nodes (LangGraph-like concept).
  - Event bus + trace store (observability).
- **Core Agents (roles)**
  - **Planner**: converts goals into task graphs (DoE generation, hypothesis trees).
  - **Design Agent**: proposes material formulation/process changes (composition, dopants, print & sinter parameters).
  - **Simulation Agent**: Live Sinter interface for distortion & densification prediction; updates compensation and schedules. Desktop Metal
  - **Printer Cell Agents**: Shop, P-1, X25 Pro, X160 Pro connectors via Live Suite; queue mgmt., job templating, telemetry. Desktop Metal
  - **Post-Process Agent**: sintering, depowdering, infiltration, heat-treat control.
  - **Metrology Agent**: property measurement pipeline (magnetic, electrochemical, density, porosity, microstructure).
  - **Critic/Verifier Agent**: checks plan steps & results against constraints & confidence (reflection patterns).
  - **Data Steward Agent**: lineage, schema enforcement, PII/IP policies.

- **Knowledge & Memory**
    - **Semantic memory (RAG)** across papers, internal notes, MSDS, vendor guides; **episodic memory** of every trial; **procedural memory** of validated "skills".
    - Retrieval and memory design are first-class concerns in agent systems. Stanford University
- **Safety & Guardrails Layer**
    - Policy engine, typed contracts, budget/timebox watchdogs, simulated dry-runs, approval gates.
- **Experiment DB & Lakehouse**
    - Stores **Experiment**, **Sample**, **Process**, **Measurement**, **Decision** records with lineage.

3.2 Physical/Integration View

- **Desktop Metal Integration**
    - **Live Suite** for fleet management, build prep (Live Build), and sintering simulation (Live Sinter) with cloud connectivity; **streaming feedback** for closed-loop control. Desktop Metal+1
    - X-Series (e.g., X25 Pro) binder-jet process advancements (e.g., TurboFuse binder for speed/strength) are captured in the tool registry as capabilities/constraints for planning & scheduling. PIM International magazine
- **Robotics & Manipulation (road-map)**
    - Robotic pick-and-place, depowdering, and fixture ops driven by language-to-action value maps and code-mediated grounding (no brittle motion primitive lists).
      VoxPoser- Composable 3D Value M…

---

4) Data Model & Contracts (typed, agentic-ready)

All inter-agent messages and tool calls are **schema-constrained**. Examples:

4.1 Core Records

```json
// ObjectiveSpec
{
  "id": "OBJ-2025-000134",
  "domain": "magnetics|electrolytes|ferromagnets|custom",
  "target_metrics": [
    {"name": "BH_max", "unit": "kJ/m^3", "goal": ">=", "value": 300},
    {"name": "Coercivity", "unit": "kA/m", "goal": ">=", "value": 900}
  ],
  "risk_class": "R3",  // requires HITL approvals
  "constraints": {"HRE_free": true, "max_cost_$kg": 150}
}
```

```json
// CandidateDesign
{
  "id": "CD-...","objective_id": "OBJ-...","provenance": {...},
  "composition": {"elements": {"Fe": 0.76, "B": 0.18, "C": 0.02, "Co": 0.04}},
  "powder_batch": "PB-...",
  "process_parameters": {
    "printer": "X25Pro",
    "layer_thickness_um": 50,
    "binder": {"type": "TurboFuse", "saturation_pct": 85},
    "sinter_schedule": {"profile_id": "SS-...", "atmosphere": "H2/N2",
"hold_steps": [...]}
  },
  "evidence": {"simulations": [...], "rationales": ["..."]},
  "safety": {"hazards": ["H2"], "mitigations": ["leak-test", "purge"] }
}
```

```json
// PrintJobSpec (to Live Suite connector)
{
  "job_id":"J-...","printer":"X25Pro","build_box_mm":[400,250,250],

"parts":[{"cad":"s3://.../part.stl","quantity":6,"shrink_compensation":{...}}],
  "powder":"316L","layer_thickness_um":50,"binder":"TurboFuse",
```

```
  "traceability":{"powder_batch":"PB-...","lot":"LOT-..."}
}
```

4.2 Agent Contracts (PEC)

```json
// PlanStep (Planner → Executor)
{
  "step_id":"PS-...","type":"print|sinter|measure|simulate|analyze",
  "inputs": {...typed...},
  "acceptance": {"metrics":[{"name":"density","op":">=","value":0.97}]},
  "safety_class":"S2","budget_tokens":12000,"timeout_s":900
}
```

```json
// Critique (Critic → Orchestrator)
{
  "step_id":"PS-...","confidence":0.62,
  "issues":["acceptance-not-met","plan-incoherent"],
  "recommended_actions":["replan-single-branch","increase-hold-2-by-10min"],
  "reflection":"Why previous assumption likely failed ... "
}
```

Rationale: The **Agent Design Pattern Catalogue** highlights reflection variants, tool registries, planning modes, and guardrails—each maps to typed contracts enabling safe orchestration. [ar5iv](ar5iv)

---

5) Core Components (detailed)

5.1 Orchestrator (Graph Runtime)

- Executes **task graphs**; persists **checkpoints**; supports **retries with backoff**, **idempotent tool calls**, **compensation actions**.
- **Budget & timebox** governance (token, cost, wall-clock).
- **Trace store** (per-node prompts, tool I/O, metrics).

5.2 Planner Agent

- Implements **Planning pattern** with **structured outputs** (JSON schemas) and **iterative re-planning** on feedback. TECHCOMMUNITY.MICROSOFT.COM
- Switchable **single-/multi-path** modes; **one-shot vs incremental** querying based on risk/cost. ar5iv

5.3 Design Agent (Materials)

- Uses **LLM-guided modification** → **validation** → **self-reflection** loop; records *modification history* to improve convergence efficiency (observed to reduce steps across test cases).
  LLMatDesign: Autonomous Materials Discovery with Large Language Models.pdf
- Tools: crystal/structure generation, MLFF relaxations, property predictors, and DFT queue (optional final).
  LLMatDesign: Autonomous Materials Discovery with Large Language Models.pdf

5.4 Simulation Agent

- Interfaces **Live Sinter** to compute distortion/compensation and temperature profiles, closing the loop to print and sinter specs. Desktop Metal

5.5 Printer Cell Agents (Shop, P-1, X25 Pro, X160 Pro)

- Live Suite connectors for **job submission, telemetry, and queueing**; map job specs and constraints to machine profiles; return **layer-wise & run telemetry** for analysis. Desktop Metal

5.6 Post-Process Agent

- Sintering/depowdering automation; parameter search + guardrails; **HITL approval** for new schedules (R3+).

5.7 Metrology Agent

- Magnetic (e.g., **BH_max**, Hc), density/porosity (Archimedes/CT), microstructure (SEM/EDS image pipelines), EIS for electrolytes; adapters emit **Measurement** records with uncertainty and calibration data.

5.8 Critic/Verifier Agent

- Applies **Self/Cross/Human Reflection**; checks constraint satisfaction, schema validity, physical plausibility; triggers **dry-run in simulation**, downgrades to **single-path**, or escalates to **HITL**. ar5iv

5.9 Data & Memory Services

- **Semantic memory (RAG)** + **episodic memory** of all runs; **procedural skills** library (validated recipes, templates).

- Emphasizes long-term retention and agent context across workflows (common in agent frameworks). Stanford University

5.10 Safety & Guardrails

- **Multimodal guardrails** on prompts, parameters, and outputs; **policy checks**; **kill-switch/timeout**; **printed-part risk class** gating. ar5iv

---

6) End-to-End Workflows

6.1 Discovery Loop (near-term: data transfer + feedback; mid-term: semi-automated; long-term: full robotics)

1. **Objective ingest** → **Planner** emits DoE graph (branching where helpful).

2. **Design Agent** proposes candidate compositions/process settings with *hypotheses* (typed JSON). **Self-reflection** added to memory each iteration.

LLMatDesign: Autonomous Materia…

3. **Simulation Agent** runs Live Sinter; computes compensation and sinter schedules; updates acceptance criteria. Desktop Metal

4. **Printer Cell Agent** submits build via Live Suite → collects telemetry for traceability. Desktop Metal

5. **Post-Process Agent** executes sinter/depowder; metrology pipeline captures measurements and uncertainties.

6. **Critic** validates outcomes vs. objective; logs **reflection** and either: (a) accepts and archives; (b) requests **re-plan** (change composition or process); or (c) escalates to **HITL**. ar5iv

6.2 Representative Application Pipelines

- **HRE-free Permanent Magnets**
  - Composition search in Fe-B-C-Co classes; explore microalloying & grain refinement; target **BH_max** and coercivity; **sinter profiles** tuned for grain boundary phases.
  - Pattern choices: **multi-path planning** early; **cross-reflection** later for safety/consistency.
- **Ferromagnets (soft/hard)**
  - Rapid screening for saturation magnetization & coercivity; print parameter sweeps (layer thickness, saturation %, binder type) with Live Sinter compensation.
- **Solid-state Na-ion electrolytes** (e.g., NASICON-like)
  - Composition space exploration for high σ; porosity/density control via binder-jet parameters + sinter profiles; EIS measurement ingestion and **critic** verification.

Why these loops? Materials agents that iterate **modify → validate → reflect → next** have shown efficiency benefits versus historyless baselines.

LLMatDesign: Autonomous Materia…

7) Reliability, Safety, and Governance

- **Typed function‑calling & strict schemas** for all tools (printers, furnaces, metrology). TECHCOMMUNITY.MICROSOFT.COM
- **Guardrails**: input range checks (e.g., binder saturation), unit sanity, powder/binder compatibility, atmosphere safety (H2), and equipment interlocks. ar5iv
- **Uncertainty‑aware gating**: require HITL if model confidence low or measurements borderline. (LLMs can be overconfident/miscalibrated in chemistry; do not trust raw self‑confidence.)
- **Audit & lineage**: immutable run logs, prompts, tool I/O, and data signatures for IP and regulatory needs.
- **Incident recovery**: node‑level retries, circuit‑breakers, **fail‑closed** behavior on any schema/policy violation, auto‑rollback plans.

---

8) Observability & Evaluation

- **Traces & Metrics** per agent step: token/cost, latency, success, schema errors, plan depth, branch efficiency, material KPIs (density, BH_max, σ).
- **Offline replay** for incident analysis; **A/B** runs to compare patterns (e.g., self‑reflection on/off).
- **Scientific LLM validation**: evaluate combinations with domain tools—an established best practice for LLMs in science.

---

9) Data & Knowledge

- **Lakehouse** (Parquet/Delta) + **Vector store** (papers, SOPs, materials DBs) with **RAG**.
- **Materials ontologies** for compositions, phases, processes, and properties; **knowledge graph** edges link *Design → Process → Measurements → Decisions*.

- **Policy-aware retrieval**: mask proprietary data per project.

---

10) Interfaces & APIs

- **Tool/Agent Registry API**
  - `GET /registry/tools` → capabilities, schemas, SLA, safety class.
  - `POST /job/print` (DM Live Suite adapter) → returns job handle, telemetry stream URL. Desktop Metal
- **Orchestrator API**
  - `POST /plan` (objective → plan graph)
  - `POST /execute` (graph id)
  - `GET /trace/{run_id}`
- **Data APIs**
  - `POST /measurement` (typed measurement with calibration & uncertainty)
  - `GET /experiment/{id}` (complete lineage)

---

11) Security & Compliance

- **AuthN/Z**: service accounts per cell; scoped tokens; signed plans.
- **Network zones**: production printers on restricted VLAN; brokered by gateway services.
- **Data governance**: project-scoped storage; encryption at rest/in flight; export controls checks on compositions.

---

12) Phased Delivery Roadmap

Phase 0 — Foundations (8–10 weeks)

- Orchestrator skeleton (graph runtime), registry MVP, schemas, Live Suite read/write + Live Sinter simulation, Experiment DB, Observability baseline. Desktop Metal+1

Phase 1 — Closed-loop Discovery (10–14 weeks)

- Planner, Design Agent (composition/process suggestions), Critic with self-reflection, Metrology ingestion, acceptance tests on **representative samples** for the three application areas.
 LLMatDesign: Autonomous Materia…

Phase 2 — Process Automation & Optimization (12–16 weeks)

- Parameter optimization (DoE + bandits), multi-path planning, cross-reflection, proactive scheduling across Shop/P-1/X25/X160 cells, Live Sinter-in-the-loop compensation. ar5iv+1

Phase 3 — Full Robotic Handling (road-map)

- Robotic part handling and depowdering; value-map-based manipulation with code-mediated grounding (VLM + LLM).
 VoxPoser- Composable 3D Value M…

---

13) KPIs & Acceptance

- **Discovery throughput**: cycles/day; time-to-first-viable candidate.
- **Quality**: % runs meeting acceptance criteria (density, BH_max, σ).
- **Agent reliability**: plan success rate, schema violation rate, mean recovery time.
- **Cost**: tokens/decision; machine utilization; cost per accepted candidate.
- **Safety**: number of guardrail interventions; zero unapproved high-risk actions.

14) Risks & Mitigations

- **LLM hallucination / overconfidence** → enforce typed schemas, reflection & cross-checks, HITL gates for high-risk steps.
  Are large language models super…
- **Data drift / simulation–reality gap** → continuously recalibrate Live Sinter and property predictors with fresh measurements. Desktop Metal
- **Integration brittleness** → contract tests and **dry-run** simulators; fallbacks to safe defaults (pause/notify).
- **IP leakage** → project-scoped RAG and strict retrieval filters.

---

15) Appendix A — Pattern ↔ Platform Mapping

Pattern (source)	Where it lives in ARC
Planning (structured outputs, iterative)	Planner Agent; JSON plan schemas; Orchestrator re-planning. TECHCOMMUNITY.MICROSOFT.COM
Tool-Use (typed tools)	Registry + typed function calls for DM printers, furnaces, metrology. ar5iv
Self-Reflection	Design & Critic Agents annotate each step; improves convergence. LLMatDesign: Autonomous Materia…
Cross-Reflection	Peer model/agent critiques plan & results before execution. ar5iv
Human Reflection (HITL)	Approval gates at risk classes R3+ (new sinter recipe, binder changes). ar5iv
One-shot vs Incremental Querying	Planner mode selection for cost vs certainty. ar5iv

Single‑ vs Multi‑path Planning	Exploration (multi-path) early → exploitation (single-path) late. ar5iv
Tool/Agent Registry	ARC Registry service; capabilities, schemas, SLAs, safety class. ar5iv
Multimodal Guardrails	Policy engine + validators on text, numbers, files, images. ar5iv
Memory (episodic/semantic/procedural)	Lakehouse + vector store + skills repo used by agents. Stanford University

---

16) Appendix B — Representative Test Sets (initial validation)

- **HRE‑free magnets**: 12 compositions across Fe‑B‑(C/Co) with print/sinter sweeps; acceptance on **BH_max**, Hc, density.
- **Ferromagnets**: Fe‑Co‑based soft magnets; saturation M_s and coercivity maps versus process parameters.
- **Na‑ion electrolytes**: NASICON‑type variants; acceptance on room‑temp ionic conductivity and densification.

---

17) Appendix C — References (select)

- Agentic patterns overview (reflection, planning, tool-use, multi-agent). Stanford University
- Pattern catalogue (self/cross/human reflection; single/multi-path; one-shot/incremental; tool/agent registry; guardrails). ar5iv
- Planning pattern with structured outputs & orchestration. TECHCOMMUNITY.MICROSOFT.COM
- Desktop Metal Live Suite + Live Sinter (cloud hub, build prep, simulation, telemetry). Desktop Metal+1

- LLM‑driven materials design with self‑reflection & modification history.
- Robotics grounding via LLM‑composed 3D value maps and code interface (for future robotic handling).
 VoxPoser- Composable 3D Value M…
- LLMs in science: combine LLMs with domain tools; exercise caution/verification.
- Overconfidence/miscalibration risks in chemistry tasks → add guards/HITL.
- Public reporting on the Agentic Design Patterns book (breadth ~21 patterns; composition). PPC Land

---

Closing note

This revision converts the original spec into a **pattern‑driven system**: every critical capability (planning, tool use, reflection, memory, safety) is bound to a design pattern with concrete implementation hooks into DM's software stack and ARC's lab infrastructure. It gives us a scalable foundation to progress from **data‑connected closed loops** to **fully autonomous print–sinter–measure cycles**—with the right safety and governance to make it production‑grade.

Fully Automated Materials Discovery Platform – Technical Figures

System Architecture Overview

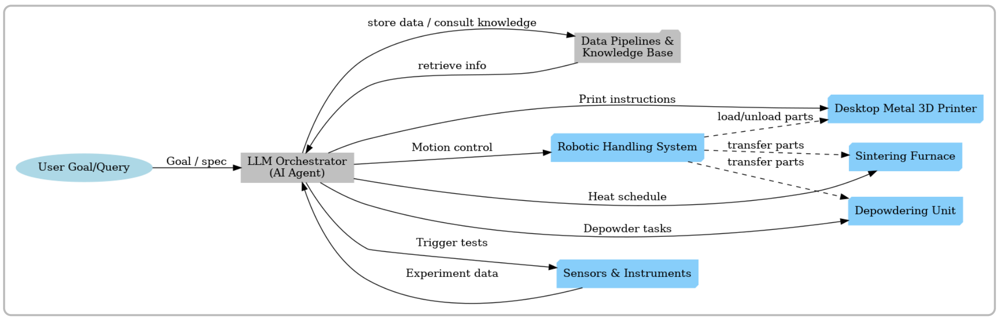

Figure 1: *High-level system architecture of the AI-driven materials discovery platform.* An LLM-based Orchestrator (AI agent) interfaces with all hardware and software layers. It accepts user goals (e.g. target material properties) and coordinates the **Physical Devices** – including a Desktop Metal 3D printer for sample fabrication, a sintering furnace for post-processing, a depowdering station for powder removal, and a robotic handling system that transfers samples between equipment. The orchestrator also communicates with **sensors and instruments** for real-time measurements, and with data/knowledge bases to log results and retrieve domain knowledge. Solid arrows denote command/control flows (e.g. printing instructions, motion commands, experiment triggers), while dashed arrows denote data flows (e.g. experimental results back to the AI, or the robot physically moving parts between devices). This architecture ensures a closed-loop integration of intelligent decision-making with automated synthesis and characterization hardware, enabling autonomous end-to-end experimentation.

Closed-Loop Experimentation Cycle

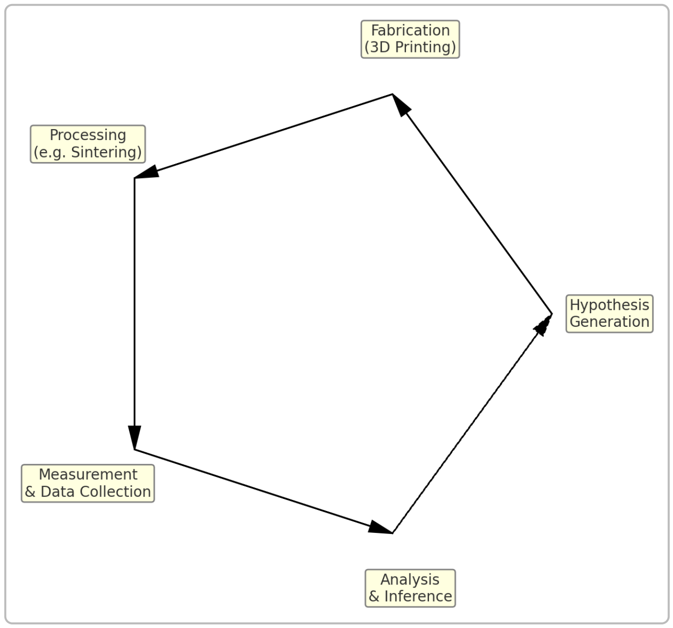

Figure 2: *Closed-loop AI-guided experimentation cycle for materials discovery.* The platform iteratively executes the scientific method in a loop of **hypothesis-driven experimentation** [1] [2]. In each cycle, the AI generates a **hypothesis** or candidate material composition/structure predicted to meet the target property. The proposed material is then **fabricated** (e.g. 3D printed) and **processed** (such as sintering in a furnace or other post-treatments). Next, the system **measures** the relevant properties of the sample using integrated sensors/instruments, collecting data on performance. The results are **analyzed** by the AI, which compares measured outcomes to target metrics. Finally, a **feedback** step (dashed arrow) closes the loop – the AI agent reflects on the outcome and uses it to inform the next hypothesis in an iterative refinement process [3] [4]. This closed-loop cycle continues until the material's performance meets the design criteria, illustrating the self-driving lab approach to materials R&D.

Physical Robotics Layout

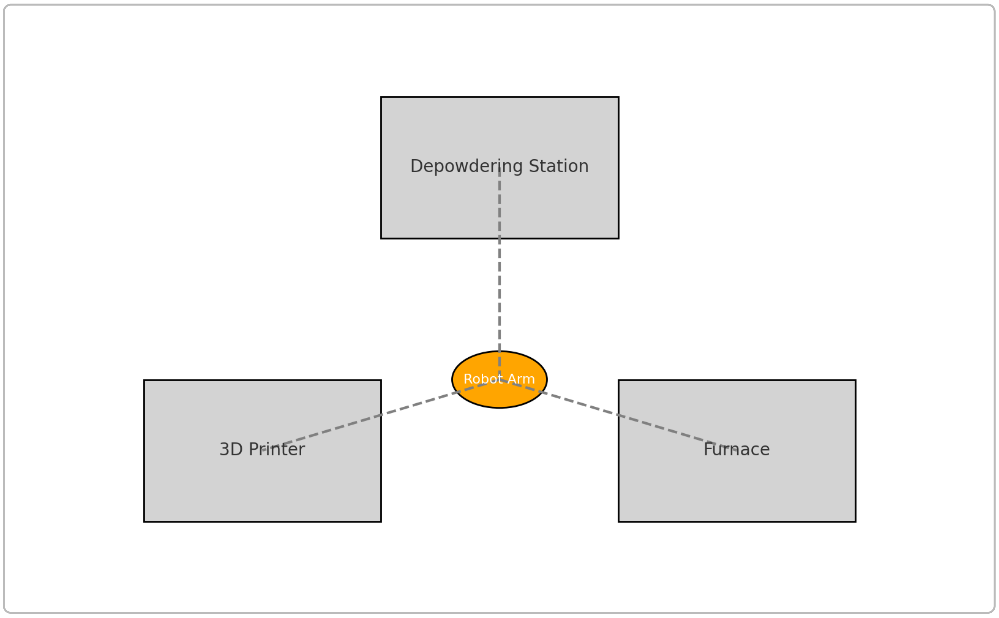

Figure 3: *Physical layout of robotic fabrication and processing components in the lab.* Key hardware elements are arranged to enable seamless robotic transfer of samples. A **Desktop Metal 3D Printer** (left) produces metal or ceramic samples from AI-suggested designs. A **Sintering Furnace** (right) is used to densify or heat-treat printed parts at high temperature. A **Depowdering Station** (top) allows for removal of loose powder and cleaning of parts (especially for binder jet or powder-bed processes). At the center, a multi-axis **robotic arm** (orange) serves as the material handling system. The robot can **reach all devices** (dashed lines indicate transfer paths), moving print jobs from the printer to the furnace or depowdering unit, and subsequently to measurement stations. This physical cell design ensures fully automated end-to-end workflows – the robot fetches a printed sample, prepares it through depowdering, places it in the furnace for sintering, and eventually delivers it to sensors for characterization – all without human intervention.

AI Decision Loop and Tool Invocation Flow

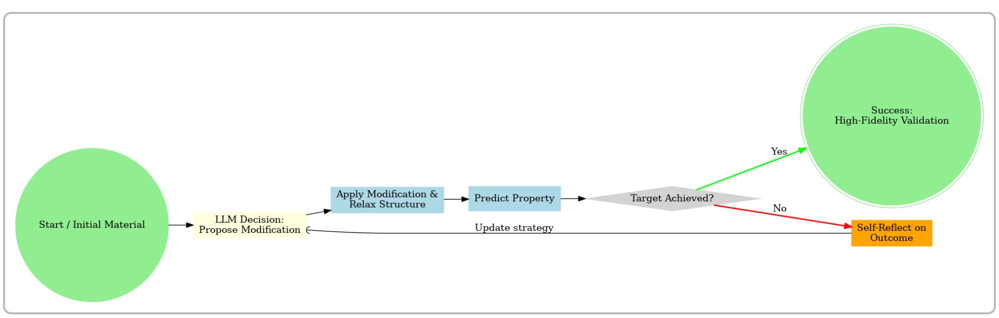

Figure 4: *AI decision-making loop with integrated simulation and self-refinement (adapted from LLMatDesign [1]).* Initially, the AI agent starts with a **base material (start)** and a target property goal. The **LLM Orchestrator** then **proposes a modification** to the material's composition or structure (e.g. substituting an element or changing microstructure) based on prior knowledge and experimental history [5] [4]. This proposed design is executed by applying the modification and running a **structure relaxation** tool (e.g. an

ML force field) to predict the stable atomic structure, followed by a **property prediction** (using a surrogate model to estimate if the modified material's property y is close to the target) [6] [7]. A decision point checks **if the target is achieved**. If yes, the cycle ends with a successful candidate (and possibly triggers a high-fidelity validation like DFT or an experiment) – indicating the material design goal is met. If no, the agent enters a **self-reflection** phase: the LLM analyzes the outcome, evaluates why the modification fell short, and **learns from this feedback** [3] [8]. It then updates its strategy (incorporating the accumulated modification history and reasoning) and iterates the loop with a new hypothesis. This closed decision loop – propose → simulate → evaluate → reflect – enables autonomous convergence toward an optimal material solution, continually invoking the appropriate computational tools and adjusting plans based on prior results.

Language-to-Action Robotics Control Mapping

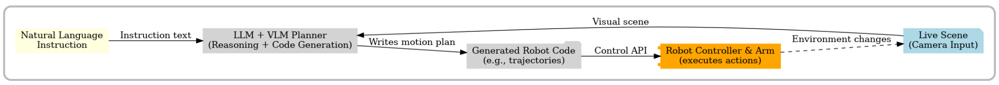

Figure 5: *Language-driven robotic control flow, integrating vision and action (inspired by VoxPoser and ELLMER frameworks [9] [10]).* The system enables high-level instructions to be translated into physical robotic actions. A **user's natural-language query** (e.g. *"prepare a sample and test its conductivity"*) is fed to the AI orchestrator. Simultaneously, a **live scene image** from the lab's camera provides the visual context of the environment (locations of equipment, objects, etc.). The platform's planner – a combination of an LLM and a Vision-Language Model (VLM) – parses the instruction and **analyzes the scene** to formulate a detailed action plan [11] [12]. The LLM+VLM planner may generate code (e.g. Python/ROS instructions or machine code for devices) representing a sequence of robot manipulations or equipment commands needed to fulfill the task. This **generated robot code** (motion trajectories, tool commands, etc.) is then sent via the control API to the **robot controller**, which executes the actions in the physical world. As the robot and devices act (printing, moving parts, adjusting knobs, etc.), the **environment updates** (e.g. a part is moved or a drawer is opened), which is detected by the camera – creating a feedback loop where the vision input informs the AI if further actions are required [13] [14]. This language-to-action mapping allows the system to go from an abstract instruction to concrete operations on hardware, essentially making **"conversation the command"** for the robotic lab.

Use Case Workflow Diagrams

The following flowcharts illustrate simplified closed-loop experimental workflows for three research use cases supported by the platform. Each diagram highlights how the general AI-driven cycle is tailored to specific goals and metrics of the project.

Use Case 1: Heavy Rare-Earth-Free Permanent Magnets

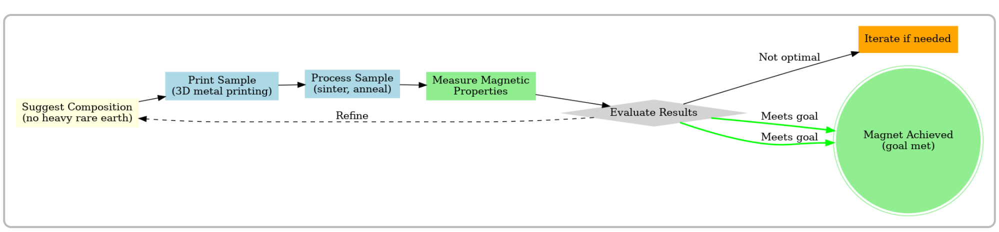

Figure 6a: *Closed-loop discovery of high-performance permanent magnets without heavy rare earth elements.* The AI suggests a new **alloy composition** for a magnet that avoids heavy rare-earth additives (leveraging knowledge of magnetic materials). That composition is then **3D printed** into a test magnet sample (e.g. via bound metal deposition or binder jet printing) and **sintered/annealed** to develop the requisite microstructure. The platform then **measures the magnetic properties** of the sample – e.g. saturation magnetization, coercivity, BH energy product – using integrated magnetometers or hysteresis loop tracers. The AI **evaluates the results** against target specifications (e.g. a certain energy product comparable to NdFeB magnets). If the magnet's performance is not yet optimal (perhaps coercivity is too low), the AI refines its hypothesis – adjusting the alloy recipe or process parameters (indicated by the dashed "refine" feedback loop) – and a new iteration begins. This loop of proposing composition → printing → processing → measuring continues until a **magnet meeting the performance goal** is achieved. Once the **target is met**, the cycle ends with a successful rare-earth-free magnet composition identified.

Use Case 2: Advanced Soft Ferromagnets

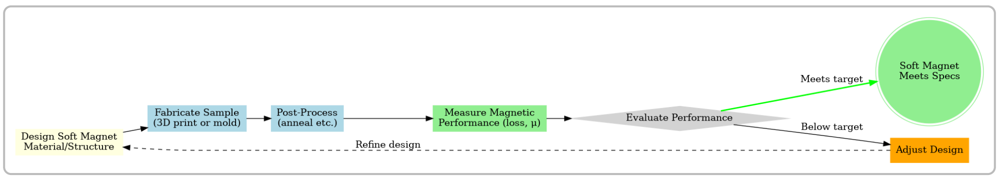

Figure 6b: *Autonomous development of advanced soft ferromagnetic materials (e.g. for efficient transformers or inductors).* The AI formulates a **soft magnet material design** – which may involve a specific Fe-based alloy or composite aimed at high magnetic permeability and low core losses. A prototype is **fabricated** (either 3D printed or cast, depending on design) and then **post-processed** (for example, thermal annealing to optimize grain structure and reduce coercivity). The sample's **magnetic performance** is then measured: key metrics include magnetic permeability (μ), saturation magnetization, and losses at operating frequency. The AI agent **evaluates the performance data** relative to the desired targets (for instance, a certain permeability and loss threshold for power applications). If the soft magnet does not meet specs – say the losses are too high – the AI will **adjust the design** (modify composition or processing, such as adding grain refiners or changing annealing temperature) and repeat the cycle. Through successive iterations, the system converges on a material that **meets the specifications** for soft magnetic behavior (green path), achieving the required combination of high permeability and low hysteresis loss.

Use Case 3: Solid-State Sodium-Ion Electrolytes

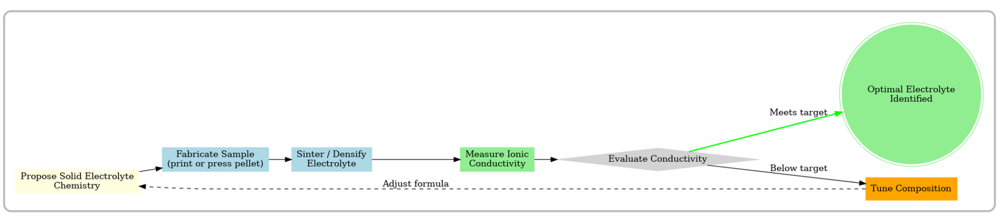

Figure 6c: *Closed-loop optimization of a solid-state electrolyte for Na-ion batteries.* The AI proposes a **solid electrolyte formulation** (a ceramic or glassy material composition that conducts Na^+ ions) based on known chemistries (e.g. NASICON-type, sulfides, etc.). Using the automated lab, a sample electrolyte is **fabricated** – for instance, 3D printing or pressing a pellet of the material – and then **sintered/densified** to attain the necessary density and phase. The platform then **measures the ionic conductivity** of the sample (e.g. via electrochemical impedance spectroscopy) along with other properties like electrochemical stability. The AI

evaluates the conductivity results against the target (for example, aiming for conductivity comparable to liquid electrolytes, on the order of 10^{-3} S/cm). If performance is below the target, the AI enters a refinement loop – **tuning the composition** (perhaps adjusting dopant levels or trying a different crystal structure) and then re-fabricating and testing. This iterative optimization continues until an **optimal electrolyte** is identified that meets the conductivity and stability requirements. The final output is a candidate solid-state electrolyte composition with high Na-ion conductivity achieved with minimal experimental cycles, demonstrating accelerated materials discovery through autonomous experimentation.

---

[1] [3] [4] [5] [6] [7] [8] LLMatDesign: Autonomous Materials Discovery with Large Language Models.pdf
file://file-7fubNpHyWE8R9GxGw31r65

[2] Hypothesis-driven closed-loop learning. Diagram showing how iterative... | Download Scientific Diagram
https://www.researchgate.net/figure/Hypothesis-driven-closed-loop-learning-Diagram-showing-how-iterative-cycles-of_fig1_40867995

[9] [12] VoxPoser- Composable 3D Value Maps for Robotic Manipulation with Language Models.pdf
file://file-WRfK1rFaxsNNt2xDoRBKUT

[10] [11] [13] [14] Embodied large language models enable robots to complete complex tasks in unpredictable environments | Nature Machine Intelligence
https://www.nature.com/articles/s42256-025-01005-x?error=cookies_not_supported&code=0ce18f54-f03d-4cf0-9521-5a4e8599be95