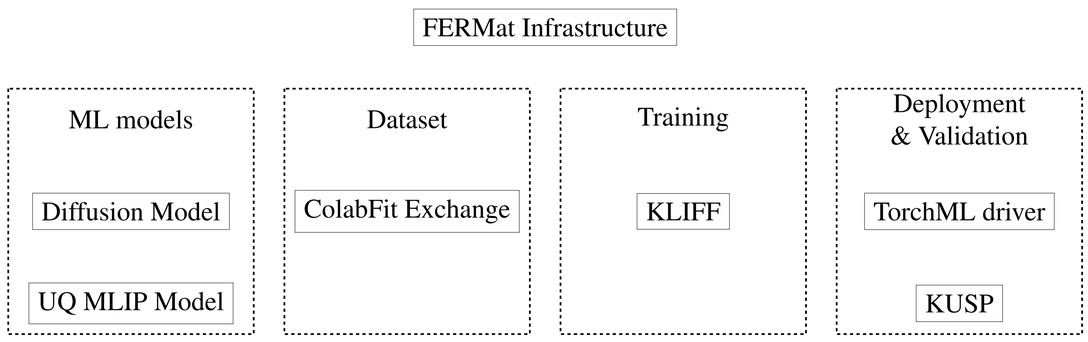
FERMat: Foundational Representation of Materials

NSF Award #2311632 *GOALI: Frameworks: At-Scale Heterogeneous Data based Adaptive Development Platform for Machine-Learning Models for Material and Chemical Discovery*

Stefano Martiniani[*], Ellad B. Tadmor[†], George Karypis[†], Adrian E. Roitberg[‡], Richard G. Henning[‡], Mingjie Liu[‡], Mark K. Transtrum[+]

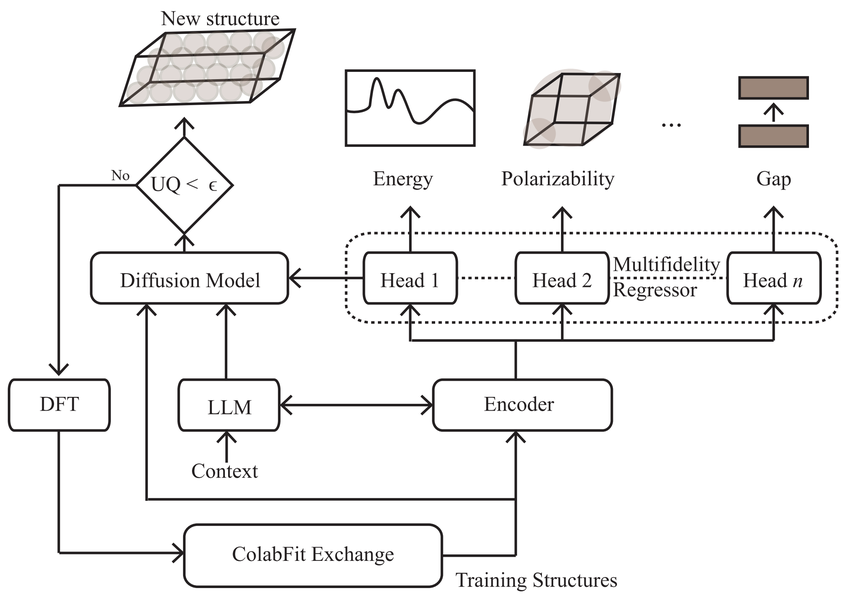
[*]New York University, [†]University of Minnesota, [‡]University of Florida [+]Brigham Young University

Introduction

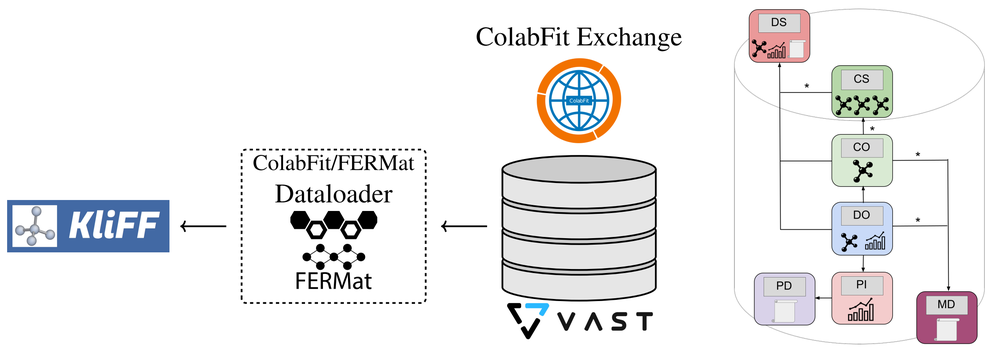


ML Models

- Self-augmenting loop of diffusion model and universal regressor
- Multifidelity property prediction with uncertainty quantification
- Context encoded by LLM



ColabFit Exchange



ColabFit dataloader schematic, along with the ColabFit data standard. (*) indicates optional relations.

Abbrv. DS: Dataset, CS: Configuration Set, CO: Configuration, DO:Data Object, PD: Property Definition, PI: Property Instance, and MD: Metadata.

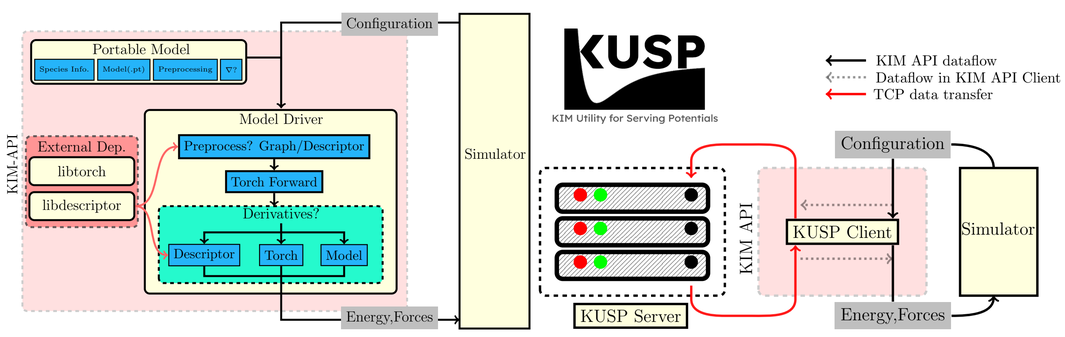
- Largest curated collection of training data **over 370 datasets** using ColabFit data standard[1]
- Highly performant *Vast Database* backend
- An on-the-fly streaming dataloader to link ColabFit to KLIFF, and Intel Open MatSci ML workflows

Training

KLIFF[2] is a high performance Python package to train all KIM API based models.

- Supports Pytorch Lightning for multi-GPU distributed training
- LMDB based large dataset handling capabilities
- Inbuilt support for ColabFit data-streaming
- Support for compiler based auto-differentiated (Enzyme) descriptor library, libdescriptor[3]

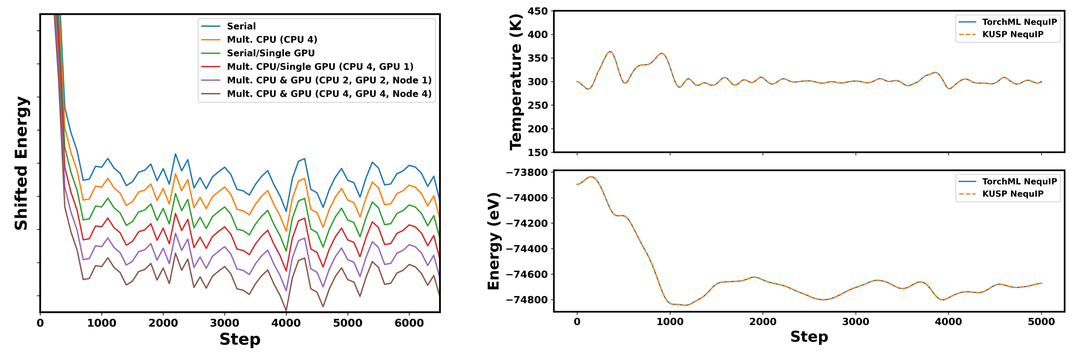
Deployment & Validation



- Uses OpenKIM suite of tests and verification checks, including upcoming Crystal Genome Framework for testing against arbitrary crystal polytype[4,5]
- KIM API based TorchML[6] model driver for driving parallel MD simulations
- Server-Client design based KUSP[7] for deploying arbitrary models (e.g. can deploy Intel Open MatSci ML models directly and test using OpenKIM)

Results

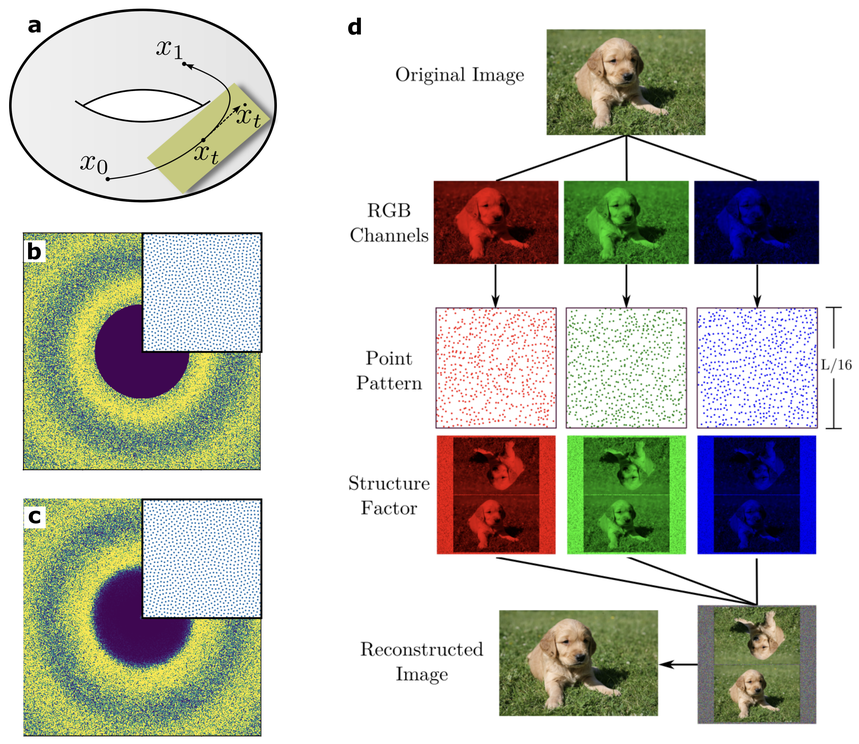
✓ ColabFit: Dataset streaming
✓ KUSP: Can run tests using both Pytorch and JAX based ML models
✓ TorchML driver: Distributed descriptor and graph convolution models



TorchML driver running NequIP potential over variety of shared and distributed CPU-GPU architectures (left), and comparison of NequIP model deployed via KUSP vs the TorchML driver (right).

Generative Model

✓ First proof that a Riemannian flow-matching model (**a**) using a graph neural network can learn an ensemble of states with a given correlation structure:
  – Model trained on stealthy hyperuniform point patterns (with suppressed structure factor for small wave vectors) generated by FReSCO (**b**).
  – Model generates structures with similar structure factor under explicit consideration of periodic boundary conditions (**c**).
- Extension to multiple particle types in point patterns by considering RGB channels of images (**d**).
  – Interpretable quality of the generated samples.
  – Robust test for generating samples from joint distributions as necessary for de-novo generation of crystalline materials that considers fractional coordinates, atom types, and lattice vectors.



- Development of open-source generative framework matching current state-of-the-art MatterGen (closed-source):
  – Inclusion of classifier-free guidance for guiding materials towards desired properties/behavior, *e.g.*, superconductivity, elastic behavior, *etc.*
  – Integration of the general stochastic interpolant framework that, among others, incorporates score-based diffusion and flow-matching methods.

Conclusion

- Bootstrapping of the diffusion model and universal multifidelity regressor
- OpenKIM suite powered distributed pipeline for large model training and validation
- Flow matching based diffusion model for material generation
- Ensemble averaged UQ NequIP-like property predictor

---

1. `colabfit.org`, **2.** `kliff.readthedocs.io`, **3.** `libdescriptor.readthedocs.io`,
4. `openkim.org`, **5.** `kim-tools.readthedocs.io`, **6.** `openkim.org/id/TorchML__MD_173118614730_000`,
7. `kusp.readthedocs.io`,