

New issue



# Release 0.6.0 #3134

Open

Assignees



Labels

- CI
- bug
- configuration
- dataset
- dependencies
- documentation
- enhancement
- evaluation
- examples
- performance
- policies
- processor
- question
- rl
- robots
- sensors
- simulation
- teleoperators
- tests
- training
- visualization



imstevempwork opened on Mar 11 · edited by imstevempwork

Edits

Collaborator








## LeRobot Community Roadmap 0.6.0

Welcome to the LeRobot Community Roadmap for v0.6.0! This document outlines what the core LeRobot team is actively working on and highlights the most impactful areas where the community can jump in to help.




How to read and use this roadmap:

- Team vs. Community: Items prefixed with [Team] are actively being tackled by the internal team. Unmarked items are prime opportunities for community contributions.
- High-Impact Contributions: The open community tasks are not just entry-level issues—they are significant, high-impact features. While the core team will eventually address them if needed, your contributions speed up our collective progress tremendously!
- Reference Links & Existing PRs: Any linked PRs or issues are provided for context and do not necessarily represent the definitive final implementation. If a task already has an open PR, jumping in to review, test, and validate it provides immense value to the project.
- Flexibility: The robotics AI field moves incredibly fast. We reserve the flexibility to pivot and change directions as the landscape evolves.
- Priority: The items listed below are not in any specific order of priority.

### 1. Dataset Infrastructure at Scale

1. Scalable dataset loader with parquet/video sharding alignment. Align parquet and video file sharding one-to-one. Fix the Dataset.num\_shards bug. Add an explicit file manifest. Optimize the streaming data loader to batch-read parquet instead of row-by-row. ()
  2. info.json as typed dataclass. Replace the free-form dict with a validated dataclass. Add explicit fields for action representation type, robot type, action/observation specs, and file manifest.
  3. Dataset slicing support. Allow dataset[start:end] indexing on LeRobotDataset. (Community issue  [Datasets don't support slicing 🙄 #3195](#), open PR  [feat\(dataset\): add efficient slice support to LeRobotDataset. \\_\\_getitem\\_\\_ #3212](#))
  4. Dataset partial downloads. Download datasets by episode, task, or tag instead of the full dataset.
  5. Episode-scale tags. Add tagging support at the episode level for filtering and organization.
- 
6. [Team] Streaming-first dataset redesign. Design a next-gen format where continuous streams are the primitive and episodes are a derived view. Support append-friendly ingestion, near-zero init cost, lazy metadata.
  7. [Team] MultiLeRobotDataset enhancement&fix. Add weighted sampling, explicit key mapping per dataset, auto-padding for absent keys, per-robot normalization stats, and custom per-sample preprocessing. Ungate TrainPipelineConfig for list-of-repo-ids.
  8. [Team] Fix floating-point drift in video timestamps. Accumulated floating-point errors during dataset concatenation cause frame lookup failures. (Community issue  [Cumulative floating-point drift in from\\_timestamp causes FrameTimestampError on concatenated videos #3177](#), open PR  [fix\(datasets\): prevent floating-point drift in video timestamps for concatenated videos #3239](#))
  9. [Team] Torchvision transforms registration. Allow users to register standard torchvision transforms in the dataset pipeline at training time. (Community open PR  [feat\(dataset\): registering torchvision transforms #3153](#))
  10. [Team] Chunk-based uploads for continuous deployments. Incremental dataset uploads that avoid full reprocessing for 24/7 rollout scenarios.

## 2. Simulation Environments, Evaluation, Benchmarks, and Leaderboard

1. [Team] Integrate 11+ simulation benchmarks for systematic VLA evaluation: SimpleEnv, RoboCerebra, Robocasa, VLAbench, Libero-Mem, Genesis AI, etc. (Community issue  [Proposal: vla-eval integration — 11+ sim benchmarks for lerobot policies #3187](#) open PR  [RoboCasa365 integration #3217](#))
2. [Team] Benchmark leaderboard. Add default training commands per model. Full eval matrix as a public leaderboard in HF.
3. [Team] Homogenize policy and evaluation API contracts. Define explicit interfaces for "LeRobot-compatible" policies, environments, and robots. Extract implicit contracts into a formal specification.
4. [Team] Hub-native debugging/evaluation loop. Upload policy → remote evaluation → leaderboard.
5. [Team] Libero nightly regression CI. Run Libero benchmarks nightly to detect policy regressions automatically. (community issue  [Can NOT reproduce the result of Gr00t-n1.5 on Libero #3228](#) *GR00T reproducibility*)

## 3. Policy Training, Verification, and Performance

1. Configurable attention mechanism. Allow choosing attention implementations (e.g., flash attention) across all models.
  2. Systematic policy verification. Verify all shipped policies train to expected metrics on reference datasets. Create a reference results table. Focus on VLA policies (SmolVLA, Pi0, Pi0-Fast, XVLA, Gr00t) which are highest-traffic.
  3. Gradient checkpointing and torch.compile for all policies. Currently only Pi0 and XVLA have gradient checkpointing; only Pi0 has torch.compile. Extend to all policies. (*open PR* 🗑️ [feat\(act\): add torch.compile support for ACT policy #3061](#) ACT torch.compile)
  4. Homogenize action tokenization. Unify how different policies tokenize actions.
  5. Homogenize freezing VLM and vision encoders. Consistent API for freezing/unfreezing across all VLA policies.
  6. Low-VRAM evaluation path. Process-isolated evaluation for GPUs ≤8GB, enabling users with consumer hardware to run VLA policies. (*Community issue* 🕒 [libero/smolvla: add low-VRAM 2-process evaluation path for ≤8GB GPUs #3098](#), *open PR* 🗑️ [feat\(eval\): add process-isolated evaluation for low-VRAM GPUs #3235](#))
  7. Optimize observation preprocessing for real-time inference. Reduce CPU-GPU synchronization overhead in the observation pipeline. (*Community open PRs* 🗑️ [perf: optimize observation preprocessing for real-time inference #3135](#), 🗑️ [perf\(processor\): reduce GPU-CPU sync in action tokenization #3123](#))
- 

8. [Team] E2E test focus and coverage reporting. Shift from unit tests toward end-to-end tests.
9. [Team] ONNX/TensorRT export for policies. Enable model export for deployment on edge devices. (*Community issue* 🕒 [Feature: ONNX / TensorRT export for trained policies #3146](#))
10. [Team] Features manipulation rework. Make policies independent of specific feature names using feature types. Fix rename\_map override issue. Consolidate the different representations of features across the codebase.

#### 4. SW Architecture, Configuration System, Developer Experience and Code Quality

1. Draccus pain point mitigation. Fix error messages, Literal type handling, and add a --help formatter for nested config fields.
  2. Typed codebase expansion (8 → 14 modules). Enable mypy strict checking for all modules. Replace dict[str, Any] with TypedDict structures.
  3. Robot and teleoperator typing. Tighter generics and TypedDict-based structures for return types. Features properties should have documented shapes.
  4. Structured logging. Rework to consistent logging.getLogger(\_\_name\_\_) pattern across all modules. Some modules use print, some use logging — needs standardization.
  5. Fix policy.path YAML config loading. Loading policies from saved YAML configs is broken. (*Community issue* 🕒 [policy.path option broken when loading config from yaml #2957](#))
-

6. [Team] Minimal default install (functionalities-only). Create pip install lerobot with the bare minimum. Training, robot control, dataset and policies become optional extras: 🔄 [refactor\(deps\): minimal default install with facade pattern and dependency isolation #3362](#)
7. [Team] Async inference refactor. Fix behavioral divergence between async and standard inference in real/simulation environments. (Community issues 🕒 [Async inference behaves differently between migrated older models and newly trained models #2980](#), 🕒 [Async inference for simulation \(libero benchmark\) #2670](#))
8. [Team] Replace unsafe pickle. The async inference module uses pickle deserialization which is an RCE vulnerability. (Community issue 🕒 [Security: Unsafe pickle deserialization in async inference enables Remote Code Execution \(CWE-502\) #3047](#), open PR 🛠️ [Fix: Replace unsafe pickle with safetensors + JSON in async inference #3048](#))
9. [Team] Action interpolation pipeline step. Interpolate actions from training frequency (30Hz) to robot execution frequency (100Hz) for smoother motion.
10. [Team] Explore torchcodec full migration (replace PyAV). Complete migration for all video decoding paths. Investigate GPU decoding and parallel decoding. Video decoding is the suspected cause of RAM spikes during training.
11. [Team] OfflineTrainer/OnlineTrainer split. Split the monolithic training script into distinct trainer classes for imitation learning vs RL.
12. [Team] Pipeline refactor: embed into robot/teleoperator. Rewrite pipelines to live inside robot and teleoperator objects. Remove the aggregate feature from policy pipelines. Simplify the ProcessorStep hierarchy.





## 5. Hardware, Sensors, and Robot UI

1. Implement more robots, cameras, motors and HW devices.
2. Camera backend robustness. Fix OpenCV camera backend failures on Linux, add V4L2 auto-selection, and improve RealSense connection timeouts. (Community issues 🕒 [opencvCameraConfig settings fail under backend=ANY on Linux; the same settings work with backend=V4L2 #3198](#), 🕒 [Read, Store and Training Adaptation of Depth Information for Stereo Camera #3070](#); open PRs 🛠️ [fix\(camera\): auto-select V4L2 backend on Linux for /dev/video\\* paths #3208](#), 🛠️ [fix D405 RealSense connection timeout on startup #3164](#), 🛠️ [feat\(camera\): add manual exposure, gain, and white balance options for RealSense cameras #3220](#))
3. SO101/Lekiwi calibration fixes. Fix wrist\_roll calibration and motor configuration (use\_degrees default). (Community issues 🕒 [so101 wrist\\_roll calibration broken in new version: leader/follower zero points don't correspond #3193](#), 🕒 [Lekiwi wrist\\_roll motor not work properly due to use\\_degrees option after v0.4.4 #3176](#))







- 
4. [Team] General robot UI. Single web-based UI covering: robot connection, camera setup/preview, teleop control, dataset recording, and upload. (open PR 🛠️ [Lerobot UI #3162](#))
  5. [Team] New sensor support (depth, force, tactile). Generic interface for recording from additional sensor types and storing in LeRobotDataset. (Community open PR 🛠️ [Add tactile sensor module with unified 6D point cloud interface #3175](#) tactile)
  6. [Team] Audio sensor support. Add microphone recording and storage in the dataset pipeline.

7. [Team] Multi-frequency modality recording (research investigation). Research whether LeRobot should natively support multi-rate sensor data. Output: written assessment, not code.

## 6. Reinforcement Learning and Research Policies

1. Training-time RTC and cloud inference. Implement RTC training integration. Explore cloud inference for large models. Publish a Hub tutorial.
- 
2. [Team] RL stack refactoring. Comprehensive refactor of the RL infrastructure. (*Community open PR*  [RL stack refactoring #3075](#))
  3. [Team] World models (VlaJEPa, DreamZero). Integrate world model policies as first candidates.
  4. [Team] Reward models (Top-Reward, Robometer). Learned reward signals beyond SARM. (*open PR*  [Reward models refactor #3142](#))
  5. [Team]  $\pi$ 0.6 RECAP RL implementation. New RL policy contribution from the community. (*Community open PR*  [feat\(rl\):  \$\pi\$ \\*0.6 RECAP implementation #3245](#))
  6. [Team] gr00t N1.7 VLA policy. Next-generation NVIDIA VLA integration. (*Community open PR*  [feat\(policies\): add NVIDIA gr00t N1.6 VLA policy #3085](#))

## 7. Documentation and Resources

1. API documentation and docstrings. Add comprehensive docstrings (Google convention) to all public APIs. Set up automatic doc testing (like transformers). Generate API reference docs.
  2. ROS2 integration exploration. Explore native compatibility with ROS2 for broader ecosystem reach.
  3. Compute hardware guide. Community-requested guide for hardware requirements by policy and dataset size. (*Community issue*  [\[Request\] Compute HW Guide/Benchmark #3191](#))
  4. Internationalization. Community-driven translation of documentation. (*Community issue*   [\[i18n-KO\] Translating docs to Korean #3058](#))
- 
5. [Team] Containerized demo environment. Curated environment for easy onboarding without dependency management.
  6. [Team] Fix and maintain examples. Multiple examples broken with recent API changes (Rerun integration, quick start dimension mismatches). (*Community issues*  [API Examples in Lerobot Imitation Learning Documentation, does not Initiate rerun.io #3219](#),  [ERROR white screen in Rerun for "lerobot-record" #3218](#),  [pi05\\_base pretrained model has action/state dimension mismatch with lerobot/libero dataset, causing official Quick Start example to fail #2963](#))
  7. [Team] Model cards for all policies on HF Hub.



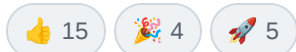
**imstevenpmwork** pinned this issue on Mar 11



SlenderMongoose on Mar 13 · edited by SlenderMongoose

Edits ▾ ⋮

I'd suggest integrating GR00T n1.6 and World Action Models (WAM).



fracapuano on Mar 21

Contributor ⋮

Ciao [@imstevenpmwork](#) 🙌 How about porting to the library [the orca hand](#)?  
For context, the orca hand is [a fully open-source](#), cheap, fully 3d-printable robot hand that can be integrated on OpenArms for fully dexterous manipulation.



Bombardelli on Mar 22 · edited by Bombardelli

Edits ▾ ⋮

I would suggest having a better way to run inference. Currently is way too heavy compared to other platform that use ROS.

Also, the implementation of Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models would be great to have inside lerobot as well as the last models published by Physical Intelligence:

Precise Manipulation with Efficient Online RL - <https://www.pi.website/download/rlt.pdf>

$\pi^*0.6$ : a VLA that Learns from Experience - <https://www.pi.website/download/pistar06.pdf>

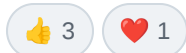


fracapuano on Mar 23

Contributor ⋮

last models published by Physical Intelligence:  $\pi^*0.6$ : a VLA that Learns from Experience

🙌 Would happily help you [@uljad](#) in porting it to the library if you're down



Ivjonok last month

⋮

I have recently discovered that realsense camera usage relies on auto-exposure and can vary from run to run. Moreover the issue is elevated when the lightning has frequency similar to fps which makes frames shutter.

One fix I found is to adjust the exposure during camera setup. It is not really invasive change, but can drastically improve reproducibility of the setup.

[#3220](#)

Happy to discuss more and adjust PR.



Grigorij-Dudnik 3 weeks ago



Most important - to make client-server work good, without glitches, and to make it support newer models - same as for normal (record script) inference.

Also, if it's possible, to make Pi0,5 load faster then for 2 minutes on beginning of client-server operation.



thunder-007 3 weeks ago



Gemini Robotics ER-1.5



fracapitano 3 weeks ago

Contributor



Gemini Robotics ER-1.5

I am afraid Gemini is not open source, though a reproduction would be so interesting 🤔



**github-actions** added

bug

CI

configuration

dataset

dependencies

documentation

3 weeks ago

50 remaining items

Load more



**github-actions** added

teleoperators

tests

training


visualization

3 weeks ago



**imstevenpmwork** self-assigned this 3 weeks ago

  **jjolla93** mentioned this [3 weeks ago](#)

 [refactor\(datasets\): replace untyped dict with typed DatasetInfo dataclass #3289](#)

 **jjolla93** 3 weeks ago

I've opened a PR for item 1.2 (info.json as typed dataclass). [#3289](#)

 2


  **jjolla93** mentioned this [2 weeks ago](#)

 [chore\(logging\): replace logging.basicConfig with init\\_logging #3297](#)

 **jjolla93** 2 weeks ago

I also opened a PR for item 4.4 (structured logging). Since logging needs to be updated across more than 100 files, I'm splitting the work into multiple PRs grouped by current logging pattern.

[#3297](#)

 **ZhongChongWang** 2 weeks ago

Hi, I'm the author of [#3193](#). I'd like to work on the SO101 calibration fix (item 5.3).

My proposed fix: bring wrist\_roll back into the calibration flow with a guided reference position step, so that leader and follower zero points correspond correctly.

I have a real SO-101 setup to test on. I'm still learning the codebase, so if anyone wants to collaborate on this fix, I'd welcome the help — I can handle the hardware testing side while someone more familiar with the code helps with the implementation.

Could someone confirm this direction is correct before we start?


  **s1lent4gnt** added  [2 weeks ago](#)

  **whats2000** mentioned this in 2 pull requests [5 days ago](#)

 [Add Depth Image to LeRobotDataset #2604](#)

 [feat\(datasets\): add depth image support \(rebase of #2604 on v3.0\) #3402](#)

  **1wos** mentioned this [3 days ago](#)

 [\[i18n-KO\] Translate bring\\_your\\_own\\_policies.mdx #3383](#)



1wos 3 days ago



Progress update on Internationalization (Section 7, item 4 → [#3058](#)).

The Korean track is active — below are PRs with translation and peer review both completed !

- [#3126](#) — Initial ko/ scaffolding + 6 translated docs (index, installation, pi0, smolvla, act, il\_robots)
- [#3371](#) — rtc.mdx
- [#3325](#) — so101.mdx
- [#3383](#) — bring\_your\_own\_policies.mdx

Dashboard — real-time tracker across Korean, Simplified Chinese, and Traditional Chinese tracks

(per-file status, outdated detection, contributor filtering): <https://translation-tracker-iota.vercel.app/>

Planning to post a consolidated update here every ~2 weeks.



[Sign up for free](#) to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

### Metadata

#### Assignees

**imstevenpmwork**

#### Labels

- CI
- bug
- configuration
- dataset
- dependencies
- documentation
- enhancement
- evaluation
- examples
- performance
- policies
- processor
- question
- rl
- robots
- sensors
- simulation
- teleoperators
- tests
- training
- visualization

#### Type

No type

#### Fields

[Give feedback](#)

No fields configured for issues without a type.

#### Projects

No projects

#### Milestone

No milestone

### Relationships

None yet

---

### Development

No branches or pull requests

---

### Participants

