A Replayable Runtime for Brownian Nanomachine Simulation

Motor Physics, Selective Tracing, and Cloud Portability

Anonymous author

15 November 2026

Abstract

Thermal fluctuations and nonequilibrium driving determine the behavior of nanoscale motors and engines. This paper pairs a simulation runtime with two analytically checkable models: a driven rotary Brownian motor and a two-bath Brownian gyrator. Independent trajectory shards run under expiring leases. A fixed experiment specification defines seed allocation, numerical kernels, and a digest-linked sequence of chunk boundaries. The coordinator verifies the structure of submitted results and builds a manifest for independent replay. Compact records retain evidence that a trajectory can be regenerated without storing every trajectory point. The deliverable includes model calibration and construction routes, source code, tests, raw performance and physics results, and Terraform profiles for DigitalOcean, Google Compute Engine, and AWS EC2. Reproducibility is assessed at separate levels of recorded integrity, same-environment replay, and agreement with analytic observables. **Keywords.** stochastic thermodynamics, molecular motors, Brownian gyrator, reproducible simulation, provenance, cloud portability.

1 Introduction

A nanoscale engine is a stochastic dynamical system coupled to a source of free energy. Symmetry breaking, a thermal or chemical drive, and relaxation jointly determine whether a useful current survives fluctuations. A simulation should estimate that current and its uncertainty over replicates, and supply a provenance record of parameter choices, estimator assumptions, and chosen numerical resolution. Recent artificial motor experiments demonstrate several distinct mechanisms for inducing rotation and transferring useful work. A scalar angular diffusion model can test a mechanism and calibrate a coarse response, while predictive work on a particular molecule needs parameters calibrated against that molecule.

This study applies the lesson of the earlier durable WebAssembly runtime to a different workload. A single computation inside a sequential guest serialized every step through a durable checkpoint chain. Independent Brownian trajectories can execute concurrently, and their summary statistics can be combined after execution. The natural durable unit is therefore a shard of trajectories. Workers can keep numerical state in memory across integration steps, publish compact trace commitments, and replay a failed shard from its logical random stream. No fast collective fabric is required for the implemented workloads.

The contributions are a physical model pair with different independent validation routes, a concrete protocol for retry-safe publication and reproducible replay, a measured study of compact tracing overhead, and a deployable CPU worker package shared by three infrastructure profiles. The word *construction* refers here to specifying an experimental analog or molecular architecture, choosing matched calibration measurements, and giving falsifiable criteria for validation. The evidence in this work consists of simulations and code-level reproduction tests.

2 Motor physics and construction paths

At small scales, useful motion must be separated statistically from thermal fluctuations. The stationary current in a periodic potential vanishes at equilibrium, including for asymmetric potentials. Directed motion requires a free-energy source and coupling that rectifies the available transitions. Mechanical torque, two unequal thermal reservoirs, a driven chemical reaction, and photoexcitation give distinct ways to supply this energy. Stochastic thermodynamics defines heat, work and entropy production along trajectories when the baths and the mechanism of forcing are specified [1, 2]. At room temperature, $k_{\mathrm{B}}T$ is approximately 4.1 pN nm. This sets a reference scale for barriers and for the work $2\pi\tau$ done per revolution against a constant opposing torque τ. A useful model must resolve the barrier and load in units of thermal energy and preserve the sign and dimensions of the work.

2.1 Driven rotor and two-bath engine

The first benchmark is an overdamped angular coordinate with a cosine potential and a constant dimensionless net torque f. Angular displacement is accumulated on the real line while the force is periodic. This distinction preserves winding counts across the branch cut. The periodic stationary-current solution provides an independent check of long-run drift [2]. With zero tilt, the equilibrium current vanishes. With zero barrier, the drift reduces to

the free angular mobility times the net torque in physical units. The benchmark torque is imposed as part of the model. It can represent the combined action of a drive and load once calibrated. Calculating conversion efficiency additionally requires the chemical or optical energy flux that produces the drive. If the drive and load torques are independently known, useful mechanical power under a constant opposing load is that load's magnitude times the mean angular velocity in the forward direction.

A laboratory analogue uses a birefringent microscopic particle in an optical torque wrench. The angular coupling of the particle to rotating linear polarization gives a periodic restoring torque. In coordinates rotating with the polarization, rotational drag supplies a constant tilt of magnitude $\gamma_{\mathrm{rot}}|\Omega|$, where Ω is the polarization rotation rate and γ_{rot} is the rotational friction coefficient. The sign depends on the orientation convention. For a twofold angular trap, its physical period is π and one can rescale the trap phase to the benchmark's 2π period. Pedaci and collaborators implemented this type of apparatus and characterized dynamics near the loss of angular lock [3]. A calibration would identify the potential from equilibrium angle statistics, infer rotational drag from relaxation or angular diffusion, and sweep the measured rotation rate below and above the deterministic depinning threshold. Temperature, polarization angle, acquisition bandwidth, drag and any resisting load must accompany the angle traces. Angular slips and winding distributions test the effective model at finite thermal noise.

The second benchmark is a Brownian gyrator. A particle moves in a quadratic trap with stiffness matrix K, coupled to noise with different intensities along orthogonal directions. A nonzero coupling c represents stiffness axes misaligned with those bath directions. With unit mobility in dimensionless units, unequal bath temperatures and nonzero coupling yield a steady probability current even when the mean position remains zero [4]. The stationary covariance checks the distribution, while a signed-area rate checks the circulation orientation and strength. Equal bath temperatures and aligned trap axes are useful zero-circulation controls. An orientation-reversing change of trap axes reverses the sign. A finite-step estimator must be compared with its discrete-time or finite-observation-interval expectation before attributing a difference to the implementation.

Argun and collaborators realized the gyrator with a charged colloidal bead in an elliptical optical trap in water. They supplied additional directional fluctuations with a randomly varying electric field and observed probability currents with the trap and noise axes misaligned [5]. Reproducing that geometry requires calibrating the trap's two stiffness eigenvalues and the angle between the stiffness and driving axes. The trap calibrated with the electric drive off gives an equilibrium reference. With the drive on, the perturbation can be characterized through its spectrum and effective diffusion coefficients. This effective-temperature construction models directional fluctuations over the instrument bandwidth. It also consumes power in the field generator, which matters if the goal is to measure device efficiency. Lin and collaborators used an optical feedback trap to demonstrate work extraction from a Brownian gyrator subject to a load [6]. The unloaded benchmark reports signed circulation, while a load model and calibrated work readout are required to claim extracted power. These colloidal experiments operate at microscopic scales and validate the underlying stochastic mechanism. Demonstrating a fabricated molecular device requires separate evidence.

2.2 From a model to a molecular device

At molecular scale, a practical design starts with a rotor, an axle and a stator or anchor, then specifies the energy source, directional gating and a way to couple motion to a load. Recent work has demonstrated distinct implementations. Wang and collaborators incorporated catalysis-driven biaryl rotors into a cross-linked polymer gel. The rotors twisted linked polymer chains and drove contraction, and a change in the fuelling system reversed rotation and drove re-expansion [7]. This design makes mechanical transduction accessible through deformation and force measurements on the network. Liu and collaborators used chiral structural asymmetry in catalysis-driven azaindole–phenylethanoic acid rotors. Changing the stereochemical environment of catalysis changed the directional bias, illustrating why forward and reverse transition rates need to be measured separately [8]. The chemical kinetic pathways and load coupling of each device determine its correspondence to a coarse-grained rotor.

Light-driven designs use a different source of directionality. Van Beek and collaborators reported two distinct rotors in one molecule whose rotational frequencies can be biased by photochemical response. Rotor structure, solvent and the illumination conditions are model inputs for such a design [9]. In an electrically driven design, Skolaut and collaborators used a platform, upright axle and chiral rotor to demonstrate unidirectional motion in a scanning tunnelling microscope junction. The interpretation involves orbital angular-momentum transfer through the axle. It therefore requires electronic transport, surface geometry and coupling to a current, all outside the benchmark Langevin models [10]. The associated public computational dataset supplies a starting point for implementing and testing that separate model [11].

A reproducible construction study would select one of these physical architectures, acquire its structural and kinetic data, and fit a model against measured observables. A chemically driven candidate can use a state graph whose edges carry reaction rates, angular increments and fuel exchange. For a reversible isothermal transition $i \to j$ driven by fuel, local detailed balance for a specified

channel gives

$$\log \frac{k_{ij}}{k_{ji}} = \frac{\Delta\mu_{ij} - (F_j - F_i) - \tau_{\mathrm{L}}\Delta\theta_{ij}}{k_{\mathrm{B}}T}. \tag{1}$$

Here F_i is the coarse-state free energy, $\Delta\mu_{ij}$ is the chemical free energy supplied in the forward channel, and $\tau_{\mathrm{L}}\Delta\theta_{ij}$ is the work done against the load. Reversing the channel reverses those exchanges. The rate ratio constrains the bias but leaves the absolute rates undetermined. Those rates, activation barriers, the load calibration and their uncertainties must be supplied by experiments or independently validated calculations [1]. A useful prediction would be a load-dependent rotation rate with uncertainty, followed by a separate measurement where fuel or light is removed, the chirality or drive is reversed, and the measured work is compared with energy input.

Model geometry also changes what must be recorded. Abdoli and Löwen found four alternating local circulations for a particle in a ring-shaped trap with anisotropic thermal fluctuations. A global signed average can cancel even when spatially resolved currents persist [12]. Their numerical study motivates replay or additional spatial observables for that extension. The present benchmarks supply analytical targets for two specified coarse models and a route to compare future measurements with simulation. Molecular synthesis, electrical junction operation, atomistic calibration and experimental verification require dedicated studies.

3 Stochastic kernels and analytic targets

Let a radian be dimensionless, choose an energy scale E_0, and scale time by a rotational drag coefficient divided by E_0. In these units T is thermal energy and f is the effective torque. With $L = 2\pi$, the rotor evolves as

$$\mathrm{d}\theta_t = (f - U\sin\theta_t)\,\mathrm{d}t + \sqrt{2T}\,\mathrm{d}W_t. \tag{2}$$

The periodic force is derived from $V(\theta) = -U\cos\theta$, with an untilted barrier of $2U$. The simulator tracks the winding of the unwrapped angle and estimates mean angular velocity from displacement per observation time. The estimator can include initialization transients. Step-size refinement and a sufficiently long observation interval are required before comparing a short run with the invariant phase law. Euler–Maruyama approximates the discrete trajectory with Gaussian increments. For $U = 0$, the increments have the exact normal law of the free driven rotor.

Write $\Phi(x) = (V(x) - fx)/T$. The stationary Fokker–Planck current $J = (f - V')p - Tp'$ on the circle is constant. Multiplication by e^{Φ}, integration from x to $x + L$, periodicity of p, and normalization give

$$Z_f = \int_0^L \int_0^L e^{\Phi(x+s)-\Phi(x)}\,\mathrm{d}s\,\mathrm{d}x,$$
$$\bar{v} = LJ = LT\left(1 - e^{-fL/T}\right)/Z_f. \tag{3}$$

The expression has $\bar{v} = f$ for the free rotor and zero drift at $f = 0$ [2]. Direction must be measured with winding or increment records. Averaging the wrapped final phase loses this observable.

The two-bath benchmark is the linear system

$$\mathrm{d}X = -KX\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W, \tag{4}$$
$$K = \begin{pmatrix} k_x & c \\ c & k_y \end{pmatrix}, \qquad D = \mathrm{diag}(T_x, T_y).$$

Require K positive definite and both effective temperatures positive. Its stationary covariance satisfies $K\Sigma + \Sigma K = 2D$. For one timestep h, put $A = e^{-Kh}$. An exact stationary Ornstein–Uhlenbeck update is

$$X_{j+1} = AX_j + \xi_j, \qquad \xi_j \sim \mathcal{N}(0, \Sigma - A\Sigma A^{\mathsf{T}}). \tag{5}$$

The noiseless phase-space drift can be small while the stationary probability current remains nonzero. For the signed twice-area convention $x\mathrm{d}y - y\mathrm{d}x$, the population rate is

$$s_0 = \frac{2c(T_y - T_x)}{k_x + k_y}. \tag{6}$$

It follows by taking expectations of $X_1\mathrm{d}X_2 - X_2\mathrm{d}X_1$ and using the Lyapunov identity. Independent additive noises give zero quadratic cross-variation, so Itô and Stratonovich cross-area expectations agree. The discrete endpoint estimator uses $h^{-1}(x_j y_{j+1} - y_j x_{j+1})$ and compares with

$$s_h = h^{-1}\left[(\Sigma A^{\mathsf{T}})_{12} - (\Sigma A^{\mathsf{T}})_{21}\right]. \tag{7}$$

The Stokes area convention includes a factor of one half. The estimator here reports twice that area. The theoretical control $T_x = T_y$ yields zero expected signed rate. Agreement with (7) checks the sampler and estimator without mistaking finite sampling cadence for a physical change in (6) [4, 5].

4 Distributed execution and reproducibility

Experiment identity. A job specification fixes model parameters, timestep, replicate count, number of timesteps, and a logical partition into trajectory shards. Each shard receives a deterministic stream derived from the experiment seed and its logical shard identifier. The derivation makes the experiment independent of claim order and of worker names. An intentional change in shard structure or timestep defines a new experiment. The implementation uses NumPy's PCG64 bit generator

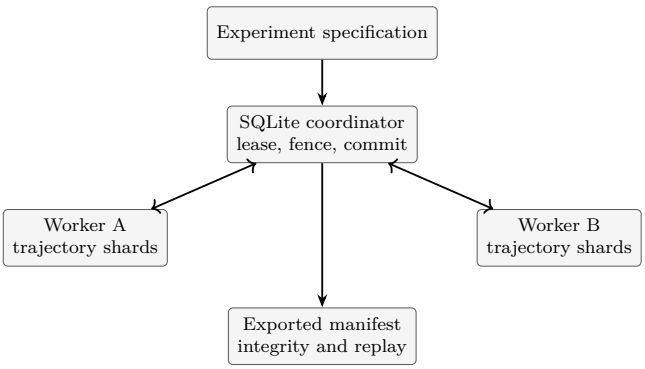


Figure 1: Workers execute separate stochastic trajectories. Only shard-level lease and publication messages cross the network. An exported manifest supports independent replay.

and normal variate sampling. Exact replay is checked in a compatible runtime environment. A fixed seed and bit generator do not imply portable floating-point output across arbitrary NumPy, CPU, or math-library builds [13, 14].

Lease and publication. The coordinator uses SQLite to persist job configurations, shard lease state, and completed shard results. The worker obtains a shard and runs its integration locally. The commit request includes a fence issued by the coordinator and a trace report. Inside the transaction that publishes a completion, the coordinator requires an active matching lease. Later attempts are excluded once the shard is committed. If a worker exits before completion, expiry makes the same logical shard eligible for another worker. In the reference runtime this replays the shard from its seed. In-memory chunks delimit trace records and do not create durable resumable state. Completed summaries are sorted by logical shard ID and combined in that order to remove the lease-completion order from floating-point aggregation.

Selective traces. A report binds chunk boundaries and compact state digests to the logical job and result. Linking the records detects accidental modifications when a verifier holds the original expected commitment. An opt-in sampled trajectory aids numerical inspection. The default records checkpoints of computational evidence rather than a trajectory dump. For S steps, R trajectories, d coordinates and a chunk stride q, a dense trajectory file contains $\Theta(dRS)$ numeric entries. A fixed-size digest per chunk adds only $\Theta(\lceil S/q \rceil)$ records per shard. Computing these digests still scans boundary state and therefore has a cost. The implementations and timing results are evaluated separately in Section 5.

What verification establishes. A verifier first checks hashes and experiment structure, and can then independently rerun a sample of shards or every shard. Full same-environment replay establishes correspondence to the reference program and declared logical inputs at the checked boundaries, conditional on the verifier trusting the comparison environment and original identity. Independent analytic or experimental evidence is required for physical validity. An auditor can pin the expected root with `-expect-root`. A hash chain without an independently retained root can be rewritten, and an operator who controls both the advertised experiment and the result can substitute a different experiment. For a future randomized audit of s uniformly chosen shards from n committed shards with b bad ones, the miss probability is

$$\Pr[\text{miss}] = \binom{n-b}{s} \Big/ \binom{n}{s}. \tag{8}$$

It applies only if samples are chosen independently after committing the manifest and replay detects every bad sampled shard. The built-in sample selector is deterministic from the root, so an adversarial audit would require independent post-commit randomness. The reference runner supplies full replay for cases requiring stronger evidence.

Proposition 1. *Under serialized coordinator transactions and fresh lease fences, at most one result is admitted for any logical shard. With a fixed compatible computation and seed mapping, the value produced by the admitted result is independent of which valid worker executes a successful retry.*

Proof. A transaction checks the shard's active fence and publishes completion, which disables future publication. Two admitted results would require the second serialized transaction to see the active lease after the first has already completed it. A retry receives the same numerical settings and logical seed, so a conforming executor in the compatible environment regenerates the same draw sequence and state transitions. The claim applies to the checked numerical implementation and excludes arbitrary worker behavior, which is addressed by replay. □

5 Validation

Analytic model checks. Each physics case used twelve logical shards of 256 trajectories with controlled seed allocation. The test compares the mean of the shard-level estimates to the analytic observable, using the between-shard standard error. The threshold is five standard errors, plus a separately stated absolute allowance of 0.015 for the Euler tilted-rotor runs. This is a reproducible numerical regression screen. Statistical variation and the permissive bias allowance limit its ability to identify weak convergence order. The tilted rotor used two observation cadences with the same simulated burn-in and measurement times. The gyrator is initialized from its invariant Gaussian, and its circulation is compared with the finite-step endpoint observable in (7). Selected

Table 1: Selected numerical observables. SE is the between-shard standard error. Observables are dimensionless.

Observable	Estimate	Oracle	SE
Free rotor velocity	0.79788	0.80000	0.00425
Free rotor diffusivity	1.02590	1.00000	0.01616
Tilted velocity, $h = .04$	0.57930	0.57992	0.00422
Tilted velocity, $h = .02$	0.58330	0.57992	0.00490
Gyrator Σ_{11}	0.80002	0.79654	0.00281
Gyrator Σ_{12}	-0.32406	-0.32468	0.00178
Gyrator Σ_{22}	1.09051	1.09740	0.00327
Gyrator s_h	0.32735	0.32819	0.00707

Table 2: Local trace policy measurement. Time is the median in ms. Bytes count the JSON trace records before the coordinator envelope.

Model	Policy	Time	Bytes
Rotor	Final digest	114.48	109
	Every 64 steps	118.21	1,296
	Plus first trajectory	118.45	33,325
Gyrator	Final digest	139.45	109
	Every 64 steps	143.01	1,296
	Plus first trajectory	138.84	49,005

estimates are in Table 1. The complete per-shard estimates and zero-circulation controls are provided in the evidence archive.

Selective trace overhead. A local x86-64 run on a host reporting an Intel Xeon Platinum 8573C used Python 3.12.14, NumPy 2.3.5, and one numerical library thread per process. Each timed shard contained 4,096 trajectories and 768 steps. After one warmup per policy, nine interleaved repetitions compared a final-only state digest, a digest every 64 steps, and the same digest schedule with all 768 points from the first trajectory. The last case samples one trajectory from the batch. An exhaustive log of all trajectories would be substantially larger. The timed kernel includes integration, statistics, and boundary hashing. HTTP, publication, and JSON conversion are outside those timings. The compact-JSON trace sizes below are computed separately from the shard's trace records. The results and final state hashes matched across all three policies.

The difference between compact and final-only median time was 3.26% for the rotor and 2.56% for the gyrator. These are comparisons at one workload shape on the local host, so they do not provide a general upper bound on overhead. In particular, the smaller median for one sampled gyrator run is within the observed timing variation. The full raw timings allow the comparison to be repeated after changing trace cadence or worker hardware.

In a second measurement, the timed code constructs the full provenance record and serializes it canonically. The compact rotor's median rose from 110.44 to 116.80 ms relative to the final-only policy, and its record grew from 867 to 3,959 bytes. The compact gyrator changed from 128.51 to 138.09 ms and from 1,220 to 4,312 bytes. The measured overheads for this layer were 5.76% and 7.45%, respectively. Publication latency and storage throughput require a deployed measurement.

Worker loss and replay. The local HTTP test used a persistent SQLite coordinator and two worker subprocesses. One of six shards was committed before restart. A different identity held the lease for shard 1 and stopped before commit. After the SQLite coordinator restarted and the lease expired, the second worker completed all six shards. The coordinator rejected the abandoned lease's attempt to submit even the otherwise identical answer with HTTP 409. Full replay matched all six records, and the verifier rejected a manifest with one changed state digest. This checks the publication protocol and independent replay for the test environment.

Local worker scaling. A second HTTP test ran one fixed 24-shard rotor workload with one, two, and four concurrently launched worker processes. It integrated 49,152 trajectories for 1,024 steps each, or 50,331,648 trajectory steps. The median times over three runs at each worker count were 2.259, 1.292, and 0.905 seconds, giving 1.75 and 2.50 times the one-worker throughput. The timed region includes worker startup, localhost HTTP, and SQLite commits. It excludes coordinator startup, submission, and export. All nine runs produced exactly the same aggregate and manifest root. The four-worker trials ranged from 0.858 to 1.572 seconds. Cluster throughput and scaling require a separate networked measurement.

6 Portability and scaling

The deliverable keeps a common worker and coordinator image and isolates provider-specific provisioning in Terraform profiles. Each profile launches one coordinator and a configurable count of execution hosts. DigitalOcean uses Droplets, Google Cloud uses Compute Engine VMs, and AWS uses EC2 instances. These providers expose different networking and instance configuration interfaces. The worker's HTTP protocol and experiment format remain the same across them. The included image build and provisioning scripts support an OCI image pinned by digest. An OCI-capable host on another provider can use the same container and worker configuration. Provider account authorization, networking policy, and an operator-supplied control credential are deployment inputs [15, 16, 17, 18].

A job with many equal-length independent shards can increase throughput by adding worker processes. If the mean shard service time is τ, there are w active workers, the coordinator processes μ lease/commit requests per second and the network can ingest B bytes per second with b result bytes per shard, a planning upper bound on

sustained completion rate is

$$\lambda \le \min\{w/\tau,\ \mu/2,\ B/b\}. \tag{9}$$

The factor of two accounts for at least one lease and one commit. Auxiliary requests and retry traffic reduce throughput. This model explains why the provider interfaces stay out of the integration loop. The dominant cost at small worker counts can be integration, while the coordinator becomes a choke point for very short shards. Large jobs can be split across coordinator namespaces with independent SQLite stores, then aggregated using stable global shard identifiers. That federation step requires a persistent index and explicit aggregation rules. The release implements a single durable coordinator.

The cloud profiles provision a small cluster whose worker count is declared explicitly. The coordinator database persists on its host across process restarts. VM loss requires a separately retained backup or durable volume. Local checks validate Terraform formatting, the provider-free module, and both rendered cloud-init roles. All three signed provider plugins initialized with lockfiles. The test environment blocked their local Unix sockets, so provider-specific schema validation remains outstanding. Live billing, worker availability, network throughput, and cross-provider bitwise equivalence require separate measurements.

7 Conclusion

Trajectory independence gives nanomachine simulation a natural unit of parallel execution. Two stochastic motor models demonstrate analytic verification of drift and circulation, and show how experimental analogs can supply parameters or an independent test of the model. Lease fencing and independently replayable shard records give a concrete path from a numerical result back to its declared inputs. Selective tracing changes what must be stored, moving cost from exhaustive trajectory I/O to sparse commitments and targeted replay. The next physical extension is to calibrate the transition and load parameters against a fabricated device or connect them to independently checked atomistic models. The first systems extension is a measured deployment on each cloud profile with the same verified job manifest.

Artifact availability and disclosure

The companion archive contains simulator and coordinator source, model specifications, tests, local evidence, deployment profiles, and this LaTeX manuscript. Instructions reproduce the numerical checks and distinguish local verification from provisioning in a paid cloud account. AI assistance supported the drafting, implementation, tests, and literature synthesis.

References

[1] U. Seifert. Stochastic thermodynamics, fluctuation theorems and molecular machines. *Reports on Progress in Physics* 75, 126001 (2012). https://doi.org/10.1088/0034-4885/75/12/126001.

[2] P. Reimann. Brownian motors: noisy transport far from equilibrium. *Physics Reports* 361, 57–265 (2002). https://doi.org/10.1016/S0370-1573(01)00081-3.

[3] F. Pedaci et al. Excitable particles in an optical torque wrench. *Nature Physics* 7, 259–264 (2011). https://doi.org/10.1038/nphys1862.

[4] R. Filliger and P. Reimann. Brownian gyrator: a minimal heat engine on the nanoscale. *Physical Review Letters* 99, 230602 (2007). https://doi.org/10.1103/PhysRevLett.99.230602.

[5] A. Argun et al. Experimental realization of a minimal microscopic heat engine. *Physical Review E* 96, 052106 (2017). https://doi.org/10.1103/PhysRevE.96.052106.

[6] W. Lin, Y.-H. Liao, P.-Y. Lai, and Y. Jun. Stochastic currents and efficiency in an autonomous heat engine. *Physical Review E* 106, L022106 (2022). https://doi.org/10.1103/PhysRevE.106.L022106.

[7] P.-L. Wang et al. Transducing chemical energy through catalysis by an artificial molecular motor. *Nature* 637, 594–600 (2025). https://doi.org/10.1038/s41586-024-08288-x.

[8] H.-K. Liu et al. Chiral catalysis-driven rotary molecular motors. *Nature Chemistry* 18, 1178–1185 (2026). https://doi.org/10.1038/s41557-025-02050-0.

[9] C. L. F. van Beek, E. Sidler, and B. L. Feringa. A photochemical rotor bias in dual molecular motors. *Nature Chemistry* 18, 1195–1202 (2026). https://doi.org/10.1038/s41557-026-02142-5.

[10] J. Skolaut et al. Unidirectional rotation in a molecular motor driven by angular-momentum transfer via helical orbitals. *Nature Chemistry* (2026). https://doi.org/10.1038/s41557-026-02264-w.

[11] S. Marek, M. Camarasa Gómez, and R. Korytár. Computational data for unidirectional rotation in a molecular motor driven by angular-momentum transfer via helical orbitals, v1.0.0. *Zenodo* (2026). https://doi.org/10.5281/zenodo.20846643.

[12] I. Abdoli and H. Löwen. Quadrupolar gyration of a Brownian particle in a confining ring. *npj Soft Matter* 2, 5 (2026). https://doi.org/10.1038/s44431-025-00015-4.

[13] NumPy. Permuted congruential generator (64-bit, PCG64). https://numpy.org/doc/stable/reference/random/bit_generators/pcg64.html. Accessed November 15, 2026.

[14] NumPy. Random compatibility policy. https://numpy.org/doc/stable/reference/random/compatibility.html. Accessed November 15, 2026.

[15] DigitalOcean. Terraform documentation for Droplets. https://docs.digitalocean.com/reference/terraform/reference/resources/droplet/. Accessed November 15, 2026.

[16] Google Cloud. Compute Engine startup scripts. https://docs.cloud.google.com/compute/docs/instances/startup-scripts. Accessed November 15, 2026.

[17] Amazon Web Services. Run commands when you launch an EC2 instance with user data input. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html. Accessed November 15, 2026.

[18] Open Container Initiative. OCI Image Format Specification. https://github.com/opencontainers/image-spec. Accessed November 15, 2026.