Dual-Mirror Operators for AI: Turning $\mathcal{F}^2$ into a Practical Invariant

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

November 1, 2025

Modern AI systems quietly rely on Fourier ideas—positional encodings, spectral convolutions, diffusion priors—yet one elementary identity is rarely treated as a design primitive: the square of the Fourier transform equals parity. Under unitary normalization, $\mathcal{F}^2$ acts as a dual mirror $R$with $Rf(x) = f(-x)$. This paper turns that textbook line into a compact, implementation-ready toolkit. With two FFTs and an index flip we obtain orthogonal projectors $P_+ = \frac{1}{2}(I + R)$ and $P_- = \frac{1}{2}(I - R)$ that separate even (mirror-consistent) and odd (mirror-reactive) content, enabling low-cost invariants, regularizers, and diagnostics across images, audio, volumes, and time series. We show how dual-mirror consistency losses improve generalization when labels are 180°-invariant; how parity-equivariant kernels reduce parameters while preserving task symmetries; how odd-leakage meters expose distribution shift and failure modes; and how fractional dual mirrors $\mathcal{F}_\alpha^2$ provide a controllable path from identity to inversion for robustness curricula and adversarial testing. The approach is architecture-agnostic, integrates cleanly with CNNs, transformers, VAEs, and diffusion models, and carries precise guarantees from harmonic analysis and metaplectic geometry. Our thesis is practical and testable: treat $\mathcal{F}^2$ as an operator, not a footnote, and you gain a portable source of invariance, interpretability, and resilience.

Keywords: dual-mirror operator, Fourier parity, parity regularization, even–odd decomposition, symmetry equivariance, fractional Fourier transform, robustness, out-of-distribution detection, diffusion models, spectral layers, SAR imaging, medical imaging, acoustics, kernel audit, metaplectic geometry.

A collaboration with GPT-5. 50 pages.

Outline

1. Abstract

   One-paragraph summary of the dual-mirror idea (R := F^2), contributions (projectors, losses, audits, curricula), domains, and results.

2. Introduction
   2.1 Motivation: from “F^2 = reflection” to an operator primitive
   2.2 Contributions and scope (what we provide, where it applies, code/overhead)
   2.3 Reader’s guide and key claims

3. Mathematical Core (copy-safe)
   3.1 Setup: unitary Fourier on L2 and extension to H^s
   3.2 Parity from F^2; order-four property F^4 = I
   3.3 Operators: R := F^2, P_plus := (I+R)/2, P_minus := (I−R)/2
   3.4 Stability: non-expansiveness in L2 and H^s; projection properties
   3.5 Commutation: translations/modulations, convolution/product automorphisms
   3.6 Hermite/eigenspaces; discrete (DFT) and torus cases
   3.7 Fractional path: F_alpha, metaplectic rotation view

4. The Dual-Mirror Toolkit
   4.1 Parity projectors (drop-in): implementations, padding, precision
   4.2 Dual-mirror consistency losses in feature/score/logit space
   4.3 Parity-equivariant layers: hard/soft kernel tying, attention/filter forms
   4.4 Even-pooling head (GEP = GAP∘P_plus)
   4.5 Audits: odd-leakage, kernel parity error, invariance score
   4.6 Local and band-limited variants; sliding-window P_plus

5. Where It Helps (and Where It Doesn’t)
   5.1 Symmetry-rich domains: SAR/ISAR, crystallography/diffraction, medical phantoms, acoustics, music/audio
   5.2 Caution zones: faces/text/road scenes, chiral cues; feature-space use and gating
   5.3 Metrics and protocols common across domains

6. Experiments (reproducible template)
   6.1 Invariance benchmarks: datasets, models (baseline vs +DM), metrics (Acc, ECE, OOD AUROC, robustness)
   6.2 Denoising & fault detection: P_plus SNR gains; real RIR asymmetry tracking via ||P_-||
   6.3 Diffusion/generative: score penalties, FID/KID, human “balance” studies

11. Conclusion

- Summary of findings, operator-centric design language, and practical takeaways.

12. References

- Indicative bibliography; venue-specific domain citations (SAR, CDI, medical, acoustics, fairness).

Appendices

- Appendix A: Minimal Spec (all building blocks)
  A.0 Assumptions; A.1 R/P_plus/P_minus; A.2 code; A.3 PyTorch snippets; A.4 fractional path; A.5 local/band-limited; A.6 padding; A.7 complexity; A.8 unit tests; A.9 GEP one-liner; A.10 training hooks; A.11 API summary.
- Appendix B: Evaluation Checklist
  B.1 symmetry checks; B.2 preprocessing; B.3 components; B.4 training; B.5 metrics; B.6 audits; B.7 ablations; B.8 compute; B.9 repro pack; B.10 fairness; B.11 pass/fail.

Abstract

Fourier structure pervades modern ML, yet a basic identity is seldom elevated to a design primitive: under unitary normalization the square of the Fourier transform equals parity. We formalize this as the dual-mirror operator $R := F^2$, which acts by reflection $Rf(x) = f(-x)$ on continuous domains (and circular index reversal on finite grids). Building on this, we present a compact, implementation-ready calculus in which two FFTs yield orthogonal projectors onto even/odd subspaces,

$P_+ = 0.5(I + R)$and $P_- = 0.5(I - R)$. These projectors enable (i) parity-aware regularizers and losses that enforce dual-mirror consistency, (ii) parity-equivariant layers via kernel tying $RKR = \pm K$, and (iii) diagnostics that quantify "odd leakage" in features and operators, exposing symmetry violations and failure modes. We extend the construction with the fractional Fourier family F_α, using F_α^2 to create a continuous path from identity to inversion for curriculum augmentation, robustness sweeps, and adversarial testing. The framework is modality-agnostic (images, audio, volumes, time series), compatible with CNNs, transformers, diffusion models, and VAEs, and carries clear guarantees from harmonic analysis (Plancherel stability, projection idempotence) and the metaplectic view (phase-space half-turns). We provide reference implementations with $O(N \log N)$ overhead, guidelines for padding and boundary handling, and evaluation protocols for symmetry-rich tasks (e.g., SAR/crystallography, room acoustics, certain medical slices). Across these settings, dual-mirror modules act as low-cost priors that improve generalization and calibration when parity is appropriate, and as transparent audits when it is not. Treating F^2as an operator rather than a footnote yields a small, portable toolkit for invariance, interpretability, and resilience in contemporary AI systems.

1. Motivation: From Textbook Fact to Operator Primitive

Most ML uses Fourier as a bridge—to move a signal into a domain where convolution is cheap, spectra are sparse, or priors are easier to express. But the composition F^2 is not merely a round trip: under unitary conventions it is the parity map. Naming this composition

$$R := F^2, P_+ := \frac{1}{2}(I + R), P_- := \frac{1}{2}(I - R)$$

and using it explicitly converts a background identity into a practical operator family that we can optimize, regularize, and audit against.

Why elevate R now?

- Symmetry shows up in real datasets. SAR targets, many crystals, certain medical slices, room impulse responses, and some musical forms are approximately invariant under a 180° flip. Parity is not exotic; it's common and exploitable.
- Cheap, composable, differentiable. $R$is "two FFTs + index reversal". That's $O(N\log\ N)$, GPU-friendly, batched, and works in any dimension.
- Operator, not augmentation. Random flips are data-level hacks. $R$and $P_{\pm}$give *algebraic control*: projections, equivariances, and tests you can reason about (idempotent, norm-nonexpansive, commutation relations).
- Bridges theory and practice. From Pontryagin duality (inversion on groups) to metaplectic geometry (phase-space half-turns), Rhas clean mathematics—useful for guarantees and proofs of stability.

What do we gain by treating $R, P_{\pm}$as first-class?

1. Invariants. Apply $P_{+}$to isolate mirror-stable content and feed it to a classifier or regressor that should be parity-blind.
2. Disentanglers. Use $P_{-}$to measure or suppress mirror-reactive components (artifacts, directed noise, asymmetries).
3. Consistency losses. Penalize feature maps or scores that break dual-mirror coherence:

$$\mathcal{L}_{\mathrm{DM}}(\phi) = \| \phi(x) - \phi(Rx) \|_2^2$$

when labels are parity-invariant; flip signs/logits where semantics invert.

4. Audits. Test learned kernels/operators Kvia $RKR = \pm K$. Quantify "odd leakage" in features

$$\rho(x) = \frac{\| P_{-}\phi(x) \|}{\| \phi(x) \|},$$

a simple health metric for drift/overfitting on symmetry-rich tasks.

Minimal implementation sketch

```
def dual_mirror(x):
    # x: (..., H, W) or any grid; use unitary FFTs
```

```
    X = fftn(x, norm='ortho')
    x2 = ifftn(X, norm='ortho')       # equals F^2 up to index convention
    return reverse_indices(x2)        # enforce exact circular reversal
```

```
P_plus  = lambda x: 0.5*(x + dual_mirror(x))
P_minus = lambda x: 0.5*(x - dual_mirror(x))
```

This generalizes to 1D/3D, complex-valued tensors, and batches. On tori/finite grids, reverse_indices is circular index reversal; on continuous domains, R is parity.

Relation to other symmetry tooling

- Data augmentation: random flips add invariance statistically; P_+ imposes it algebraically (projection), often with better calibration and less variance.
- Group-equivariant nets: powerful but bespoke per group. $R$is a *single* cheap operator available everywhere, and can coexist with equivariant layers.
- Spectral regularizers: many penalize magnitude patterns. Dual-mirror constraints incorporate phase/orientation, catching failures invisible to magnitude-only priors.

When *not* to use Rblindly

- Tasks with essential handedness (text direction, road scenes, faces, chiral molecules). Use feature-space dual-mirror (on internal $\phi(x)$), class-conditional application, or learn a gate that disables the constraint where parity is false.
- Nonperiodic boundaries: FFTs assume wraparound. Use reflection/symmetric padding or windowing before R; document the padding policy.

Generality across domains

- Gridded data (images, volumes, audio, time series): $R$is two FFTs + reversal—vectorized and stable (by Plancherel, $\| P_{\pm} f \|_2 \leq \| f \|_2$).
- Finite/torus domains: $R$is circular index reversal; $P_{\pm}$ are orthogonal projections in the DFT basis.
- General LCA groups: $R$is pullback by inversion $x \mapsto x^{-1}$; the calculus extends to cyclic groups and multi-dimensional tori without redesign.

Bottom line. Reframing "F^2 = reflection" as the dual-mirror operator Rturns a line in a textbook into a portable, differentiable primitive that yields invariants, disentanglers, losses, and audits—with theory-backed stability and near-zero engineering friction.

2. Mathematical Core

Setup and convention. Work on R^d with unitary Fourier transform
F f(w) = (2pi)^(-d/2) * integral_{R^d} f(x) e^{-i w.x} dx.
Then F is unitary on L^2(R^d), extends to tempered distributions S'(R^d), and preserves Sobolev H^s norms.

Parity from F^2. For f in S(R^d),
F^2 f(x) = f(-x) (parity)
F^4 = I (order four)
Proof sketch (distributional calculus): F^2 f(x)
= integral f(t) [integral e^{-i w.(t+x)} dw] dt
= (2pi)^d integral f(t) delta(t+x) dt
= f(-x), under unitary scaling.

Projectors and algebra.
Define R := F^2 (parity) and
P_+ := (1/2)(I + R), P_- := (1/2)(I - R).
Then
P_+^2 = P_+, P_-^2 = P_-, P_+ P_- = 0, P_+ + P_- = I,
and they are orthogonal projections on L^2. Moreover,
Range(P_+) = { even functions }, Range(P_-) = { odd functions }.

Stability (L^2 and H^s).
By unitarity of F and boundedness of R on L^2,
||P_+ f||2 <= ||f||2, ||P- f||2 <= ||f||2.
Since F: H^s -> H^s is an isometry for all s in R (via multiplier (1+|w|^2)^{s/2}),
the same bounds hold in H^s: ||P+ f||{H^s} <= ||f||{H^s}, and similarly for P_-.
Thus R, P_+, P_- are well-behaved, nonexpansive regularizers on these scales.

Commutation relations.
Let T_a f(x) = f(x-a) (translation) and M_b f(x) = e^{i b.x} f(x) (modulation).
Then
R T_a = T_{-a} R, R M_b = M_{-b} R,
P_+ T_a = (1/2)(T_a + T_{-a}) + (1/2)(T_a - T_{-a})R,

and analogously for P_-. In particular, parity flips the sign of translation and modulation parameters.

Convolution/multiplication interplay.
F(f * g) = (Ff)(Fg), F(f g) = (Ff) * (Fg).
Applying F^2 gives
R(f * g) = Rf * Rg, R(f g) = (Rf)(Rg).
Hence parity is an automorphism for both pointwise product and convolution.

Spectral picture via Hermite functions.
Let {h_n} be the Hermite basis on R (tensor products on R^d). Then
F h_n = i^n h_n, so R h_n = F^2 h_n = (-1)^n h_n.
Even degrees are P_+-eigenfunctions (eigenvalue +1), odd degrees are P_--eigenfunctions (eigenvalue -1).

Distributional scope.
All identities extend to S'(R^d). In particular, R delta_x = delta_{-x}, and R commutes with differentiation up to sign:
R(∂^alpha f) = (-1)^{|alpha|} ∂^alpha (Rf).

Fractional path (FrFT).
Let F_alpha be the fractional Fourier transform with F_0 = I, F_1 = F, F_2 = R, and
F_alpha F_beta = F_{alpha+beta} (indices modulo 4).
Interpretation: F_alpha is a rotation by angle alpha*(pi/2) in phase space; thus F_2 is the pi-rotation (parity).
A tunable “symmetry sweep” is obtained by penalties of the form
L_alpha(f) = || f - F_alpha^2 f ||_2^2,
continuously interpolating between identity (alpha=0) and parity (alpha=2).

Discrete/finite grids (DFT).
On Z_N^d with unitary DFT matrix U,
U^2 = J, where (Jx)[n] = x[(-n) mod N] (circular index reversal),
so U^4 = I. Define R := U^2, P_+ := (I+R)/2, P_- := (I-R)/2.
All projector, stability, and convolution facts carry over with circular boundary conditions.

Boundary and padding notes (implementation).
FFT implies periodic wraparound. To approximate continuous parity on finite windows, use symmetric/reflection padding or smooth tapers before applying R. Document padding in any empirical protocol.

Summary of operator facts (at a glance).

- Parity: R f(x) = f(-x), R^2 = I, R* = R.
- Projectors: P_+, P_- orthogonal, P_+ + P_- = I, R = P_+ - P_-.
- Nonexpansive: ||R f||*{H^s} = ||f||*{H^s}, ||P_± f||*{H^s} <= ||f||*{H^s}.
- Algebra: R(f*g) = (Rf)*(Rg), R(fg) = (Rf)(Rg).
- Symmetry control: F_alpha yields a continuous family from identity to parity.

3. The Dual-Mirror Toolkit

3.1 Parity Projectors (drop-in)

Goal. Extract the even/odd parts of a tensor using only FFT primitives so the operation is fast, differentiable, and batched.

Definition. With unitary FFTs, $R := F^2$ acts as a circular index reversal on finite grids. The orthogonal projectors are

$$P_+ = \frac{1}{2}(I + R) P_- = \frac{1}{2}(I - R).$$

Pseudo-implementation (N-D, batched):

```
def dual_mirror(x):
    # x: [..., D1, D2, ..., Dn], real or complex, periodic boundaries
    X  = fftn (x, norm='ortho')        # unitary FFT
    x2 = ifftn (X, norm='ortho')       # equals F^2 up to index convention
    return reverse_indices (x2)        # circular reversal along last n dims

def P_plus(x):   return 0.5 * (x + dual_mirror(x))
def P_minus(x):  return 0.5 * (x - dual_mirror(x))
```

Reverse indices. For each spatial/temporal axis of length D, map index $k \mapsto (-k) \bmod D$. In 2-D images this is the 180° wraparound flip.

Complexity. Two FFTs per call: $O(N \log N)$ with a small constant; fully vectorizable on GPU/TPU.

Differentiability. All ops are linear; gradients flow through FFTs in all major autodiff frameworks.

Numerical notes.

- Use norm='ortho' (unitary) to avoid extra scaling.
- Mixed precision works; for delicate tasks (phase retrieval), prefer float32/float64.
- For real signals, FFT/IFFT may return tiny imaginary parts; take real() if required.

Boundary handling. FFT assumes periodic wraparound. For finite-window data, prefer:

- Reflection padding (mirror pad) before applying R, or
- Smooth tapering (e.g., Tukey window) to reduce edge energy.
  Document the policy; it affects $P_{\pm}$ near borders.

When to use. Apply P_plus for symmetry-dominant signals (PSFs, room IRs, many SAR targets, some CT slices). Track $\| P_{-}x \|$ as an odd-leakage health metric for drift/faults.

3.2 Dual-Mirror Consistency Loss

Purpose. Encourage internal representations to respect parity when labels/physics are 180°-invariant.

Basic form (feature-space):

$$\mathcal{L}_{\mathrm{DM}}(\phi) \;=\; \| \phi(x) - \phi(Rx) \|_2^2.$$

Why feature-space? Many tasks aren't strictly parity-invariant at pixels but *are* at the level of class or latent structure. Enforcing consistency on $\phi(\cdot)$is safer and more flexible than on inputs.

Variants.

- Label-aware. If class semantics invert under R(rare), compose with a known permutation/sign: $\| \phi(x) - \Pi\, \phi(Rx) \|^2$.
- Head-space parity. Encourage logits $g(\cdot)$ to satisfy $g(x) \approx g(Rx)$(or $g(x) \approx \Pi g(Rx)$).
- Self-supervised. Treat (x, Rx) as positives in SimCLR/BYOL; optionally mix fractional dual-mirror (below) for a harder curriculum.
- Selective application. Gate the loss by confidence or class prior: apply $\mathcal{L}_{\mathrm{DM}}$ only when parity likely holds.

Regularization strength. Start small (e.g., $\lambda \in [0.01, 0.1]$); increase if validation under 180° rotations improves without harming base accuracy.

Practical tip. Combine with P_plus pre-pooling:

$$\mathrm{GEP}(x) := \mathrm{GAP}(P_+(x))$$

to stabilize logits while $\mathcal{L}_{\mathrm{DM}}$ shapes features.

3.3 Parity-Equivariant Layers

Objective. Constrain linear operators K(filters, attention kernels, Toeplitz blocks) to commute/anticommute with R:

$$RKR = \pm K.$$

"+" enforces even kernels; "−" enforces odd kernels.

CNN weight tying (2-D example).
Let $W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times H \times W}$.

- Even kernel: $W = \mathrm{flipud}(\mathrm{fliplr}(W))$.
- Odd kernel: $W = -\mathrm{flipud}(\mathrm{fliplr}(W))$.

Enforce either by:

1. Hard tying: parameterize only half the coefficients and mirror them on forward pass.
2. Soft penalty: add $\lambda \parallel RWR \mp W \parallel_2^2$ to the loss; anneal $\lambda \uparrow$.

Attention / linear layers.
For an operator Krepresented implicitly (e.g., FFT-domain filters), enforce

$$\widehat{K}(\omega) = \pm \widehat{K}(-\omega)$$

by averaging with its flipped version each update, or penalizing the deviation.

Benefits.

- Fewer degrees of freedom; better sample efficiency on symmetry-rich problems.
- Clear interpretability: channels split into even/odd responses.

- Compatible with standard init (He/Xavier) by constructing tied tensors post-init.

Edge cases.

- Don't force parity in early layers if input statistics are not parity-balanced; move constraints to mid/high layers or apply them class-conditionally.
- For odd kernels, beware cancellation when followed by even pooling—pair design choices deliberately (e.g., keep odd paths before non-linearities).

Short recipes (plug-and-play):

- Audit a trained model: report $\rho(x) = \| P_{-}\phi(x) \| \overline{\|\phi(x)\|}$ per batch; rising ρoften precedes accuracy drops under parity-consistent corruptions (blur/speckle).
- Denoise symmetric signals: replace x by P_plus(x) before inference; measure SNR gain.
- Parity-aware data aug: include Rxin batches; couple with $\mathcal{L}_{\mathrm{DM}}$ rather than relying on augmentation alone.
- Kernel sanity check: compute $\epsilon_{\pm}(K) = \| RKR \mp K \| / \| K \|$ for each conv block; flag high values where symmetry should hold.

4. Where It Helps (and Where It Doesn't)

4.1 Symmetry-Rich Domains

SAR / ISAR / radar.
Many point-spread signatures and target micro-Doppler patterns are invariant (or nearly so) under a 180° rotation in the image plane. Enforcing dual-mirror consistency:

- Reconstruction: add $\| x - P_{+}x \|^2$ as a proximal penalty inside iterative back-projection; reduces speckle anisotropy and improves phase stability.
- Classification: mix input augmentation $(x, Rx)$with feature-space loss $\| \phi(x) - \phi(Rx) \|^2$; parity-equivariant kernels in early/mid layers cut parameters and boost data efficiency.
- Diagnostics: odd leakage $\rho = \| P_{-}x \| \overline{\|x\|}$ flags pointing/registration errors; track ρacross passes to detect drift.

Typical gains: +1–3% accuracy on symmetric target sets; improved calibration (↓ECE) and robustness to 180° rotations.

Crystallography / diffraction (centrosymmetric lattices).
For centrosymmetric structures, the object (or its autocorrelation) is even.

- Phase retrieval: project intermediate estimates with P_+every k iterations; narrows the feasible set and accelerates convergence.
- Noise suppression: use P_+as a denoiser on Patterson maps; measure SNR gain and R-factor reduction.
- Model priors: in learned solvers (unrolled CDIs), penalize $\| RKR - K \|$ for linear blocks to keep kernels parity-consistent.

Medical imaging (selected tomographic slices).
Certain cross-sections (e.g., bilateral anatomy approximations, phantoms, coil-combined MR magnitude) benefit from parity control.

- Artifact separation: decompose reconstructions $x = P_+x + P_-x$; streaks and bias fields often concentrate in P_-x, while anatomy concentrates in P_+x.
- Regularization: inject $\lambda \| P_-x \|^2$ into variational objectives when parity prior is justified (phantoms, QA, device calibration).
- Caveat: do not enforce parity on anatomy that is naturally asymmetric; prefer feature-space DM loss on intermediate representations.

Acoustics / room impulse responses (RIRs).
Ideal direct-path IRs are close to even after appropriate centering.

- Denoise: $x \leftarrow P_+x$ before fitting parametric decay models; typically yields measurable SNR improvement and tighter T60 estimates.
- Fault detection: increases in $\| P_-x \|$ indicate occlusions, mic misplacement, or hardware faults.
- Source separation: enforce evenness on reverberation tails while allowing the early reflections to carry controlled odd components.

Music / audio (retrograde-compatible structure).
Some motifs tolerate retrograde (time-reversal) relationships.

- Generative priors: add $\| x - \lambda Rx \|$ with learned or scheduled λ to encourage self-mirroring phrases; improves perceived balance.

- Analysis: use P_+ to extract palindromic content; P_- to isolate directional gestures (pick attacks, onsets).
- FrFT curriculum: sweep α in F_α^2 to train robustness to graded time-reversal perturbations.

Metrics & protocols (common across domains).

- Invariance score: $\mathrm{Inv} = \mathbb{E} \parallel \phi(x) - \phi(Rx) \parallel \frac{}{\mathbb{E}} \parallel \phi(x) \parallel$ (↓ is better).
- Odd leakage: $\rho(x) = \parallel P_- x \parallel \frac{}{\|x} \parallel$ (track over time/slices).
- Robust accuracy: accuracy under R and under F_α^2 sweeps.
- Calibration: ECE before/after dual-mirror constraints.
- Overhead: report ms/batch added by two FFTs.

4.2 Caution Zones (and how to handle them)

Semantically oriented vision (faces, text, road scenes, chiral cues).
Parity is not a free symmetry: "b" vs "d", left vs right turn lanes, facial asymmetries.

- Use feature-space consistency, not input-space projection: apply $\mathcal{L}_{\mathrm{DM}}$ to intermediate $\phi(\cdot)$ known to encode class-invariant structure (textures, mid-level shapes), not raw pixels.
- Class-conditional gating: apply R only to labels or subsets where semantics are invariant; for others, compose with a permutation/sign flip Π (e.g., lane direction).
- Learnable gate: train a small gate $g(x) \in [0,1]$ that scales $\mathcal{L}_{\mathrm{DM}}$ and is regularized by sparsity; the model "opts in" to parity where evidence supports it.

Biomedical content with true asymmetry.
Don't enforce parity on organs with lateralization or lesions.

- Mixed policy: restrict P_+ to calibration phantoms and QA; keep clinical inference parity-free, or constrain only background/artefact channels.

Nonperiodic boundaries / small fields of view.
FFT implies wraparound; naive R can inject boundary artifacts.

- Padding/tapers: use reflection padding or windowing; document policy.

- Local parity: apply Rin overlapping windows and blend; avoids global wraparound artifacts.

Sequence models (NLP).
Word order and script direction are not parity-invariant.

- No input enforcement. If used, restrict $R$to latent spaces aligned with acoustics or prosody (for speech) or to synthetic tasks where parity is meaningful.

Non-abelian domains / irregular meshes.
DFT-based $R$is not directly applicable.

- Alternative: build parity via mesh symmetries or spectral bases on the manifold; or use graph automorphisms implementing the appropriate inversion.

Quick decision checklist

1. Does the target/label stay the same under 180° rotation or time reversal?
   If yes → consider P_+, $\mathcal{L}_{\mathrm{DM}}$, parity-equivariant kernels.
   If no → restrict to feature-space, class-conditional, or skip.
2. Are boundaries periodic or handled (pad/taper)?
   If not → fix before applying R.
3. Will enforcing parity hide rare but real asymmetric signals?
   If yes → use audits only (monitor ρ), do not constrain training.
4. Can fractional sweeps help?
   Use F_α^2 to probe robustness and find regimes where parity partially holds.

Bottom line. Apply dual-mirror tools where physics or semantics justify parity; otherwise keep them in *audit mode* (metrics and diagnostics) or constrain only those parts of the model known to be parity-compatible.

5. Experiments

Common setup (applies to 5.1–5.3)

- Framework: PyTorch (or JAX); unitary FFTs (norm='ortho').
- Padding policy: reflection padding for nonperiodic inputs; document the exact pad width per task.

- Dual-mirror ops: R(x)=reverse_indices(ifftn(fftn(x, 'ortho'), 'ortho')), P_plus=(I+R)/2, P_minus=(I-R)/2.
- Seeds & splits: 5 seeds {0,1,2,3,4}; report mean ± std. Use a fixed stratified split where applicable.
- Training: cosine LR, AdamW (lr=3e-4, wd=1e-4), batch=64 (images) / 16 (volumes) / 128 (1-D).
- Logging: store per-epoch metrics, calibration reliability diagrams, and odd-leakage histograms (||P_- x||/||x||).
- Compute overhead log: wall-clock (ms/batch) with and without dual-mirror; report GPU/CPU model.

5.1 Invariance Benchmarks

Goal. Verify that dual-mirror tools improve performance and robustness on datasets with true 180° symmetry.

Datasets (pick ≥2):

- SAR: symmetric target chips (e.g., canonical vehicles/structures cropped to square).
- Synthetic crystals: simulated diffraction or Patterson maps with exact centrosymmetry.
- Medical phantoms: mirrored CT/MR phantoms or bilateral organ phantoms (nonclinical).

Models (same depth/width across variants):

1. Baseline: CNN or ViT without parity tooling.
2. +DM loss: add feature-space $\mathcal{L}_{\text{DM}} = || \phi(x) - \phi(Rx) ||_2^2$ with weight $\lambda \in \{0.01, 0.05, 0.1\}$.
3. Parity-equivariant kernels: enforce $RKR = +K$ in early 2–3 conv blocks (hard tying).
4. Even-pooled head: replace GAP by GAP$\circ P_+$.

Input augments: standard flips/rotations + include Rx with 50% probability.

Metrics:

- Top-1 / mAP.
- Calibration: ECE (15 bins), Brier.

- OOD (180°): AUROC when test batch is replaced by Rx.
- Robustness: accuracy under parity-consistent corruptions (Gaussian blur, speckle, low-pass).
- Invariance score: $\mathrm{Inv} = \mathbb{E} \; || \; \phi(x) - \phi(Rx) \; ||_2 / \mathbb{E} \; || \; \phi(x) \; ||_2$.

Ablations:

- Apply $\mathcal{L}_{\mathrm{DM}}$ at {block2, block4, head}.
- Turn on/off even-pooling.
- Tie only the first conv vs first three convs.
- Reflection vs circular padding.

Expected table (fill with mean ± std):

Model	Acc (%)	ECE (%)	OOD AUROC	Robust@blur	InvScore
Baseline					
+ DM loss (λ=0.05)					
Parity-equiv kernels					
Even-pooled head					
DM + kernels + evenpool					

PyTorch snippet (feature-space loss):

```
def dual_mirror(x):
    X  = torch.fft.fftn(x, dim=spatial_dims, norm='ortho')
    x2 = torch.fft.ifftn(X, dim=spatial_dims, norm='ortho')
    return x2.flip(dims=spatial_dims).real


def dm_loss(phi_x):
    phi_rx = phi(dual_mirror(x))
    return ((phi_x - phi_rx)**2).mean()
```

5.2 Denoising & Fault Detection

Goal. Show P_+ increases SNR for even signals and P_- flags asymmetric faults.

Synthetic even signals:

- Generate x_{even} (e.g., even Gaussian mixtures or even PSFs).
- Noise $n$that is not even (colored noise, one-sided bursts): $y = x_{\text{even}} + n$.
- Prefilter: $\hat{x} = P_+ y$.

Metrics:

- SNR gain (dB): $\Delta\text{SNR} = 10\log_{10}(\| x \|^2/\| y - x \|^2) - 10\log_{10}(\| x \|^2/\| \hat{x} - x \|^2)$.
- MSE / PSNR / SSIM (images).
- Frequency-wise error: plot radial spectra of error before/after.

Real acoustic IRs (recorded or simulated):

- Baseline IR h_0 (centered).
- Induce asymmetry: block path, shift mic a few cm, tilt source.
- Measure $\rho = \| P_- h \| \frac{}{\|h} \|$over time/conditions; correlate with T60 estimate error and early-to-late energy ratio.

Ablations:

- Reflection vs circular padding before P_+.
- Apply P_+at raw waveform vs STFT magnitude vs cepstrum.
- Compare $P_+$with classical denoisers (median, bilateral) at matched runtime.

Expected table:

Scenario	SNR_in (dB)	SNR_out (dB)	ΔSNR (dB)	ρ_before	ρ_after
Synthetic (colored)					
Synthetic (bursty)					
RIR (blocked path)					
RIR (mic shift)					

5.3 Diffusion / Generative

Goal. Impose dual-mirror coherence on generative models where data has 180° symmetry (or where parity-balanced outputs are desirable).

Score model penalty (DDPM/score-matching):

$$\mathcal{L}_{\text{DM}}^{\text{score}} = \| s_\theta(x,t) - R\, s_\theta(Rx,t) \|_2^2.$$

Add to standard loss with weight $\lambda \in \{0.001, 0.01, 0.05\}$.

Evaluation:

- FID/KID on symmetric datasets.
- Dual-mirror consistency: $\mathbb{E} \; || x - P_+ x ||$ — | x || for generated samples.
- Human "balance" study: pairwise preference between baseline vs DM-regularized samples for perceived symmetry/balance.
- FrFT robustness: sweep αin F_α^2, measure distributional stability (e.g., MMD) under αperturbations.

Ablations:

- Apply penalty at U-Net mid-blocks vs all scales.
- Train with/without even-pooled discriminator (for GANs).
- Class-conditional switch for categories with known handedness.

Reporting table:

Model	FID ↓	KID ↓	Consistency ↓	Pref(%) ↑	Overhead (ms/batch)
DDPM baseline					
+ DM score penalty (λ)					
+ FrFT curriculum (α-scan)					

Pseudocode (penalty in training step):

```
def dm_score_penalty(score_fn, x, t):
    s1 = score_fn(x, t)
    rx = dual_mirror(x)
```

```
    s2 = score_fn(rx, t)
    return ((s1 - dual_mirror(s2))**2).mean()
```

Overhead, memory, and batching

- Runtime: measure median ms/batch over 100 steps; report separately for forward-only (inference) and forward+backward (training).
- Memory: peak VRAM with and without DM modules; FFTs are in-place in many libs but verify allocator behavior.
- Scaling: show overhead vs input size (e.g., 128^2, 256^2, 512^2).

Statistical testing & reporting

- Use paired t-tests across seeds for Acc/ECE/OOD AUROC; report p-values.
- For human studies, report n, CI (Wilson), and inter-rater agreement (Cohen's kappa).
- Release code, configs, and exact padding/windowing choices.

Failure analysis (what to look for)

- False symmetry: accuracy drop on asymmetric subsets → reduce λ, gate by class, or move DM loss to deeper features.
- Boundary artifacts: spikes at borders in P_x→ switch to reflection padding or local-window DM.
- Odd suppression: generative dullness (over-smooth) → lower λ or confine penalty to mid-frequency bands.

Repro pack checklist

- dual_mirror, P_plus, P_minus ops (N-D).
- Padding/taper utilities with fixed parameters.
- DM loss hooks (features, scores, logits).

- Kernel-tying wrappers (even_kernel, odd_kernel).
- Metrics: Acc/mAP, ECE, AUROC@R, InvScore, ρ, ΔSNR, FID/KID.
- Scripts for ablations and α-sweeps (FrFT).
- Seeded dataloaders and splits; results CSVs; plotting scripts.

This template makes the claims testable, comparable across labs, and easy to adopt with minimal code (two FFTs and an index flip).

6. Theory Notes (why it generalizes)

6.1 Bias–variance tradeoff under a true parity prior

Let $\mathcal{H}$ be your hypothesis class (e.g., a network family) and let $\mathcal{H}^{(+)} = \{h \in \boldsymbol{\mathcal{H}}: \boldsymbol{h} = \boldsymbol{h} \circ \boldsymbol{R}\}$ be the parity-invariant subclass. If the Bayes predictor $\boldsymbol{h}^{\star}$ satisfies $\boldsymbol{h}^{\star} = \boldsymbol{h}^{\star} \circ \boldsymbol{R}$ (labels invariant under 180°), projecting any $\boldsymbol{h} \in \boldsymbol{\mathcal{H}}$ to

$$\boldsymbol{\Pi}_{+}(\boldsymbol{h}) := \boldsymbol{P}_{+} \circ \boldsymbol{h} = \frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{h} + \boldsymbol{h} \circ \boldsymbol{R})$$

("Reynolds operator" for the order-2 group) yields:

- No Bayes bias: $\boldsymbol{\Pi}_{+}(\boldsymbol{h}^{\star}) = \boldsymbol{h}^{\star}$.
- Lower variance/complexity: $\boldsymbol{\Pi}_{+}$ is 1-Lipschitz in $\boldsymbol{L^2}$ (see §6.2), so any norm- or margin-based complexity $\mathfrak{R}(\cdot)$ satisfies
  $\mathfrak{R}(\boldsymbol{\Pi}_{+}(\boldsymbol{\mathcal{H}})) \leq \mathfrak{R}(\boldsymbol{\mathcal{H}})$.
- Generalization bound (sketch): for loss $\boldsymbol{\ell}$1-Lipschitz in its first argument, any PAC/Rademacher bound for $\boldsymbol{h}$transfers to $\boldsymbol{\Pi}_{+}(\boldsymbol{h})$ with no worse complexity term and (typically) smaller empirical risk on parity-balanced data.

If the prior is false (task has real handedness), $\boldsymbol{\Pi}_{+}$ introduces bias. This is why we (i) gate the constraint, (ii) apply it in feature space, or (iii) anneal its strength.

6.2 Regularization as projection (contractive, parameter-free)

$\boldsymbol{P}_{+}$and $\boldsymbol{P}_{-}$are orthogonal projections in $\boldsymbol{L^2}$:

$$\boldsymbol{P}_{\pm}^{\mathbf{2}} = \boldsymbol{P}_{\pm}, \boldsymbol{P}_{+}\boldsymbol{P}_{-} = \mathbf{0}, \langle \boldsymbol{P}_{\pm}\boldsymbol{f}, \boldsymbol{f} - \boldsymbol{P}_{\pm}\boldsymbol{f} \rangle = \mathbf{0}.$$

Immediate consequences:

- Non-expansive: $\| \boldsymbol{P}_{\pm} \boldsymbol{f} \|_2 \leq \| \boldsymbol{f} \|_2$. More generally for Sobolev norms, $\| \boldsymbol{P}_{\pm} \boldsymbol{f} \|_{H^s} \leq \| \boldsymbol{f} \|_{H^s}$.
- Proximal viewpoint: $\boldsymbol{P}_+$ solves

$$\min_{g:\, g = g \circ R} \| g - f \|_2^2,$$

i.e., the best parity-invariant approximation—no hyperparameters, no tuning.

- Spectral norm: $\| \boldsymbol{P}_{\pm} \|_{op} = 1$; hence inserting $\boldsymbol{P}_+$ in a network preserves Lipschitz upper bounds and keeps many stability proofs intact.

6.3 Group-theoretic lens (why it ports to many domains)

For a locally compact abelian (LCA) group $\boldsymbol{G}$with inversion $\iota: x \mapsto x^{-1}$,

- the Fourier transform $\boldsymbol{F}$maps $L^2(G) \to L^2(\widehat{G})$,
- double Fourier brings you back with inversion: $\boldsymbol{F}^2 \boldsymbol{f} = \boldsymbol{f} \circ \iota$ (up to convention constants),
- hence $\boldsymbol{R} := \boldsymbol{F}^2$ is pullback by inversion, and

$$\boldsymbol{P}_+ = \frac{1}{2}(\boldsymbol{I} + \boldsymbol{R})$$

is the group average over $\{\boldsymbol{I}, \iota\}$ (the Reynolds operator).

This immediately extends the toolkit to:

- Cyclic grids $\mathbb{Z}/N\mathbb{Z}$(DFT): $\boldsymbol{R}$ is circular index reversal, $\boldsymbol{P}_{\pm}$ are orthogonal in the unitary DFT inner product.
- Tori $\mathbb{T}^d$: same formulas with multi-index wraparound.
- Non-abelian outlook: via Peter–Weyl, averaging over the order-2 subgroup $\{e, \iota\}$ still yields a projection onto the $(+1)$ isotypic component (the antipode appears in the group algebra). Practical recipe: apply $\boldsymbol{P}_+$ in any domain where “inversion” is a known symmetry.

6.4 Metaplectic/phase-space picture (why $\boldsymbol{F_\alpha}$ works)

On $\mathbb{R}^{\boldsymbol{d}}$, $\boldsymbol{F}$ implements a $\boldsymbol{\pi/2}$rotation in phase space $(\boldsymbol{x}, \boldsymbol{\xi})$. Therefore:

$$\boldsymbol{F_\alpha} = \text{rotation by } \boldsymbol{\alpha} \cdot \frac{\boldsymbol{\pi}}{\boldsymbol{2}}, \boldsymbol{R} = \boldsymbol{F^2} = \text{half-turn.}$$

Consequences:

- Hermite eigenstructure: $\boldsymbol{Fh_n} = \boldsymbol{i^n h_n} \Rightarrow \boldsymbol{Rh_n} = (\boldsymbol{-1})^{\boldsymbol{n}} \boldsymbol{h_n}$. Even/odd decouple cleanly; $\boldsymbol{P_\pm}$are diagonal in this basis.
- Fractional curricula: $\boldsymbol{\alpha} \in [\boldsymbol{0}, \boldsymbol{2}]$ sweeps a continuous family from identity ($\boldsymbol{\alpha} = \boldsymbol{0}$) to parity ($\boldsymbol{\alpha} = \boldsymbol{2}$), letting you probe representation stiffness: how stable are features under partial phase-space rotations?
- Stability: metaplectic operators (incl. $\boldsymbol{F_\alpha}$) are unitary on $\boldsymbol{L^2}$ and bounded on $\boldsymbol{H^s}$; inserting them keeps norms controlled while offering a principled augmentation axis.

6.5 Algebraic commutations (what plays nicely)

Let $\boldsymbol{T_a f(x)} = \boldsymbol{f(x - a)}$ (translation) and $\boldsymbol{M_b f(x)} = \boldsymbol{e^{ib \cdot x} f(x)}$ (modulation). Then

$$\boldsymbol{RT_a} = \boldsymbol{T_{-a} R}, \boldsymbol{RM_b} = \boldsymbol{M_{-b} R},$$

so parity flips translation/modulation parameters. For convolutions and products:

$$\boldsymbol{R(f * g)} = \boldsymbol{(Rf) * (Rg)}, \boldsymbol{R(fg)} = \boldsymbol{(Rf)(Rg)},$$

i.e., $\boldsymbol{R}$ is an automorphism for both operations—why even/odd splits cleanly through linear layers and many nonlinear pipelines (pointwise activations commute with $\boldsymbol{R}$).

6.6 Data augmentation vs operator projection (why projection wins in-class)

Random 180° flips approximate invariance statistically. $\boldsymbol{P_+}$ enforces it algebraically:

$$\boldsymbol{h} \mapsto \boldsymbol{\Pi_+(h)} = \frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{(h + h \circ R)}.$$

Invariance error after projection is exactly zero (when desired), and calibration typically improves because the hypothesis set shrinks without adding loss noise. In practice, the best results often combine both: augment for diversity, project/regularize for exact symmetry.

6.7 Discrete/finite effects (Nyquist, boundaries)

- Wraparound: FFT implies periodic boundaries; $\boldsymbol{R}$ is exact on tori/grids. For finite windows, use reflection padding or tapers before $\boldsymbol{R}$ to avoid border artifacts.
- Sampling: when content is near Nyquist, 180° flips can alias subtle high-frequency asymmetries; prefer oversampling or antialiasing filters before applying $\boldsymbol{P}_{\pm}$.

6.8 When the prior is partially true (soft symmetry)

If only some frequencies/regions obey parity, make $\boldsymbol{P}_{+}$ band- or space-selective:

$$\boldsymbol{P}_{+}^{\mathcal{B}} = \boldsymbol{F}^{-1}\,\square\big(\boldsymbol{\chi}_{\mathcal{B}}(\boldsymbol{\xi})\cdot\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{I} + \boldsymbol{R})\,\hat{\cdot}\; + \;(\mathbf{1} - \boldsymbol{\chi}_{\mathcal{B}})\cdot\boldsymbol{I}\big),$$

or window $\boldsymbol{R}$ locally (overlapping patches with smooth blending). This retains gains where symmetry holds and avoids bias elsewhere.

6.9 Takeaway—why it generalizes

- It's a group-average projection (order-2), hence contractive and bias-free when the symmetry is true.
- It's unitary-friendly (metaplectic), so stability and norm control follow from standard Fourier tools.
- It's domain-agnostic on any grid/torus and extendable to general LCA groups.
- It composes cleanly with common ML layers and losses, yielding smaller effective hypothesis classes and improved robustness—at the cost of two FFTs.

7. Practical Recipes

7.1 Even-Pooling Head (drop-in, zero hyperparameters)

What. Replace global average pooling (GAP) by global even pooling (GEP):

$$\mathrm{GEP}(x) \;:=\; \mathrm{GAP}\big(P_{+}(x)\big) \;=\; \mathrm{GAP}\big(\tfrac{1}{2}(x + Rx)\big)\,.$$

This suppresses orientation-reactive activations before the classifier head.

Why. When labels are invariant under 180° flips, averaging the activation map with its dual-mirror removes odd components that destabilize logits and calibration.

How (PyTorch-like pseudo):

```
def dual_mirror(x, spatial_dims=(-2, -1)):      # NCHW by default
   X  = torch.fft.fftn(x, dim=spatial_dims, norm='ortho')
   x2 = torch.fft.ifftn(X, dim=spatial_dims, norm='ortho')
   return x2.flip(dims=spatial_dims).real


def gep_head(feat):                             # feat: [N,C,H,W]
   even_feat = 0.5 * (feat + dual_mirror(feat))
   return even_feat.mean(dim=(-2, -1))          # GAP∘P_+
```

Where to place. Use GEP at the last feature map before logits. For very asymmetric inputs, prefer feature-space GEP at a deeper block instead of raw pixels.

Gotchas.

- Non-periodic boundaries: use reflection padding inside the backbone to reduce wraparound artifacts.
- Don’t combine with odd-only kernels in the same last layer (they’ll be nulled by GEP).

Expected effect. Small accuracy lift (+0.5–2%), more stable calibration (↓ECE), and improved robustness to 180° rotations on symmetry-rich tasks. Overhead: two FFTs.

7.2 Parity-Aware Contrastive Augmentations (self-supervised or supervised)

What. Treat $(\boldsymbol{x}, \boldsymbol{Rx})$ (and optionally $(\boldsymbol{x}, \boldsymbol{F}_{\alpha}^{2}\boldsymbol{x})$) as structured augmentations.

Self-supervised (SimCLR/BYOL style).

- Positives: $(\boldsymbol{x}, \boldsymbol{Rx})$, $(\boldsymbol{x}, \boldsymbol{F}_{\alpha}^{2}\boldsymbol{x})$ for random $\boldsymbol{\alpha} \in [\mathbf{0}, \mathbf{2}]$.
- Negatives: unrelated samples; optionally include inverted-semantics pairs if your domain has classes that flip under parity.

Supervised.

- Add a dual-mirror consistency loss on embeddings or logits:

$$\mathcal{L}_{\mathbf{DM}} = \| \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{Rx}) \|_{2}^{2},$$

only for classes known to be invariant. For classes that swap under $\boldsymbol{R}$, compose with the known permutation/sign.

Curriculum with fractional path.

- Sample $\boldsymbol{\alpha}$from a schedule (e.g., uniform → anneal to $\{\mathbf{0}, \mathbf{2}\}$). This probes representation stiffness under partial phase-space rotations.

Pseudo-snippet:

```
x1 = aug(x)                  # standard crops, color jitter, etc.
x2 = dual_mirror(aug(x))     # parity pair
z1, z2 = enc(x1), enc(x2)
loss = contrastive(z1, z2) + λ_dm * ((z1 - z2)**2).mean()
```

Tips.

- Start with small λ_dm ∈ [1e-2, 1e-1].
- For imbalanced symmetry, gate the loss by a confidence score or class prior.

7.3 Kernel Tying in CNNs (even/odd equivariant filters)

What. Enforce even or odd parity on convolution kernels so layers obey $\boldsymbol{RKR} = \pm\boldsymbol{K}$.

2-D kernels (C_out, C_in, H, W):

- Even: $\boldsymbol{W} = \mathbf{flipud}(\mathbf{fliplr}(\boldsymbol{W}))$.
- Odd: $\boldsymbol{W} = -\mathbf{flipud}(\mathbf{fliplr}(\boldsymbol{W}))$.

Hard tying (parameterization).

```
def tie_even(W_half):
    # W_half is a learnable half; mirror to full W
    W_full = mirror_to_full(W_half)          # implement your indexing
    W_mirr = torch.flip(W_full, dims=(-2, -1))
    return 0.5 * (W_full + W_mirr)


def tie_odd(W_half):
    W_full = mirror_to_full(W_half)
    W_mirr = torch.flip(W_full, dims=(-2, -1))
    return 0.5 * (W_full - W_mirr)
```

Soft tying (penalty).
Add $\boldsymbol{\lambda} \parallel \boldsymbol{RWR} \mp \boldsymbol{W} \parallel_2^2$ to the loss (λ ≈ 1e-3–1e-2) and let training enforce parity.

Design choices.

- Make the first 1–3 conv blocks even for symmetry-dominant tasks; keep later layers free for nuanced asymmetries.
- Pair odd kernels with skip connections or non-even heads if you need directional sensitivity downstream.

Benefits. Fewer parameters, better data efficiency, clearer interpretability (channels separate into even/odd responses).

7.4 Phase-Retrieval Plug-in (CDI/ptychography)

What. If the specimen (or its autocorrelation) is centrosymmetric, add an explicit $\boldsymbol{P}_+$projection in the reconstruction loop.

Gerchberg–Saxton / Fienup loop (sketch).

```
for it in range(T):
    # Fourier magnitude projection
    U = fftn(x, norm='ortho')
    U = replace_magnitude(U, measured_mag)      # keep phase
    x = ifftn(U, norm='ortho')
    # Real-space constraints (support, non-negativity, etc.)
    x = real_space_constraints(x)
    # Dual-mirror projection for centrosymmetry
    x = 0.5 * (x + dual_mirror(x))              # x ← P_+(x)
```

Why it helps. $\boldsymbol{P}_+$ shrinks the feasible set to parity-consistent candidates, speeding convergence and improving stability under noise.

Notes.

- If only the Patterson map is truly centrosymmetric, apply $\boldsymbol{P}_+$there rather than on the object itself.
- Ensure correct centering/alignment; parity projection assumes the symmetry center is at the grid origin.

Metrics to report. R-factor, reconstruction MSE/PSNR, convergence speed (iterations to tolerance).

7.5 Local/Selective Dual-Mirror (partial symmetry)

What. Apply $\boldsymbol{P}_+$ locally where symmetry is expected (e.g., background, reverberation tails, crystal subregions).

Sliding-window recipe.

1. Extract overlapping windows (e.g., 64×64, 50% overlap).
2. Apply $\boldsymbol{P}_+$ per window.
3. Reconstruct with smooth overlap-add.

This prevents global wraparound artifacts and respects spatially varying symmetry.

7.6 Feature-Space Gating (avoid false priors)

What. Learn a gate $\boldsymbol{g}(\boldsymbol{x}) \in [\mathbf{0}, \mathbf{1}]$ that scales parity constraints:

$$\mathcal{L}_{\mathbf{DM}}^{\mathbf{gated}} = \boldsymbol{g}(\boldsymbol{x}) \parallel \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{R}\boldsymbol{x}) \parallel^2 \boldsymbol{\phi}_{\text{even}} = \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{P}_+(\boldsymbol{\phi}(\boldsymbol{x})) + (\mathbf{1} - \boldsymbol{g}(\boldsymbol{x})) \boldsymbol{\phi}(\boldsymbol{x}).$$

Regularize $\boldsymbol{g}$with sparsity (e.g., $\boldsymbol{L}^{\mathbf{1}}$) so the model opts into parity only where evidence supports it.

7.7 Diagnostics Pack (fast sanity checks)

- Odd-leakage meter: $\boldsymbol{\rho}(\boldsymbol{x}) = \parallel \boldsymbol{P}_{-}\boldsymbol{x} \parallel / \parallel \boldsymbol{x} \parallel$. Track per batch/epoch; spikes often signal misalignment, padding mistakes, or drift.
- Kernel audit: $\boldsymbol{\epsilon}_{\pm}(\boldsymbol{K}) = \parallel \boldsymbol{R}\boldsymbol{K}\boldsymbol{R} \mp \boldsymbol{K} \parallel_{\overline{\parallel \boldsymbol{K}}} \parallel$ block-wise; flag layers that violate expected symmetry.
- Invariance score: $\mathbf{Inv} = \mathbb{E} \parallel \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{R}\boldsymbol{x}) \parallel_{\overline{\mathbb{E}}} \parallel \boldsymbol{\phi}(\boldsymbol{x}) \parallel$ (↓ is better) for trained models.

7.8 Complexity & engineering notes

- Cost. Each use of $\boldsymbol{R}$ adds two FFTs (forward/backward), typically negligible vs conv/attention backbones.
- Precision. Prefer float32/64 for phase-sensitive tasks (phase retrieval); bf16/fp16 generally fine for classification.
- Batching. fftn over spatial dims only; vectorize across batch and channels.
- Padding. Use reflection padding or tapers for nonperiodic fields; document policy for reproducibility.

7.9 Quick “when to use what” table

Situation	Use
Symmetry-dominant classification	GEP head (+ optional DM loss)
Small data, strong symmetry	Parity-tying in early conv blocks
Denoising symmetric signals	x ← P_plus(x) prefilter
Drift/fault monitoring	Track ρ = ||P_- x|| / ||x||
CDI/ptychography with centrosymmetry	Insert P_plus in the reconstruction
Mixed/partial symmetry	Local P_plus (sliding windows)
Unsure symmetry; risk of bias	Gated DM loss in feature space

These recipes are intentionally minimal: each needs only two FFTs and an index flip, integrates with standard stacks, and comes with clear failure modes and diagnostics.

8. Limitations and Failure Modes

8.1 False symmetry priors (task not parity-invariant)

Problem. Enforcing $\boldsymbol{R}$-invariance when labels truly depend on handedness (text direction, road scenes, faces, chiral structures) injects bias and can erase discriminative cues.
Symptoms. Accuracy drop on asymmetric subclasses; increased confusion between left/right classes; overly smooth generative samples.
Mitigations.

- Class-conditional application. Only apply $\boldsymbol{P}_{+}/\ \boldsymbol{\mathcal{L}}_{\mathbf{DM}}$ to classes verified invariant; compose with a known permutation $\boldsymbol{\Pi}$ for classes that *invert* under $\boldsymbol{R}$.
- Feature-space (not input). Apply the constraint to mid/high-level features $\boldsymbol{\phi}(\boldsymbol{x})$, not to pixels, preserving low-level handedness.
- Learnable gates. Train $\boldsymbol{g}(\boldsymbol{x}) \in [\mathbf{0}, \mathbf{1}]$ to scale $\boldsymbol{\mathcal{L}}_{\mathbf{DM}}$; regularize $\boldsymbol{g}$(e.g., $\boldsymbol{L}^{\mathbf{1}}$) so the model opts into parity only where evidence supports it.
- Monitoring. Track per-class invariance score $\mathbf{Inv}_{\boldsymbol{c}} = \mathbb{E} \parallel \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{R}\boldsymbol{x}) \parallel / \mathbb{E} \parallel \boldsymbol{\phi}(\boldsymbol{x}) \parallel$. Rising $\mathbf{Inv}_{\boldsymbol{c}}$ while accuracy falls ⇒ turn parity off for class $\boldsymbol{c}$.

8.2 Boundary artifacts on nonperiodic grids

Problem. FFT assumes wraparound; applying $\boldsymbol{R}$ naively creates seams at borders.
Symptoms. Ringing near edges; spikes in $\boldsymbol{P}_{-}\boldsymbol{x}$ localized at boundaries; reconstruction drift in inverse problems.
Mitigations.

- Padding. Use reflection (mirror) padding or symmetric extension before $\boldsymbol{R}$; remove after.
- Tapers. Multiply by a smooth window (e.g., Tukey) before $\boldsymbol{R}$to reduce edge energy.
- Local $\boldsymbol{R}$. Apply $\boldsymbol{R}$ in overlapping windows with smooth overlap-add (SOA) to avoid global wraparound.
- Documentation. Fix and disclose padding/taper parameters; they are part of the reproducibility.

8.3 Partial symmetry (only some regions/bands obey parity)

Problem. Global $\boldsymbol{P}_{+}$ suppresses legitimate odd content where symmetry does not hold.
Mitigations.

- Band-selective parity. Apply $\boldsymbol{P}_{+}$ only to a frequency band $\mathcal{B}$ using masks in the Fourier domain.
- Spatially local parity. Windowed $\boldsymbol{P}_{+}$ on selected regions (backgrounds, reverberant tails) while leaving ROIs unconstrained.
- Adaptive policy. Learn masks for where/when to apply $\boldsymbol{P}_{+}$ with sparsity regularization.

8.4 Aliasing and sampling issues

Problem. Near-Nyquist content folds under the 180° flip, hiding asymmetries.
Mitigations.

- Oversample or pre-filter high frequencies.
- Report robustness at multiple sampling rates; check that $\boldsymbol{P}_{-}\boldsymbol{x}$ does not vanish purely due to aliasing.

8.5 Numerical precision and complex phases

Problem. Phase-sensitive workloads (phase retrieval, interferometry) are brittle under low precision; FFTs may introduce tiny imaginary parts for real inputs.
Mitigations.

- Prefer float32/64 for phase-critical steps; use .real after $\boldsymbol{R}$ for real-valued tasks.
- Keep normalization unitary (norm='ortho') to avoid scale creep.

8.6 Optimization pathologies

Problem. Too-strong dual-mirror penalties can underfit; odd components vanish prematurely.
Mitigations.

- Anneal $\boldsymbol{\lambda_{\mathbf{DM}}}$ from small to moderate; monitor validation under $\boldsymbol{R}$.
- Layer selection. Apply constraints in middle blocks; leave early/late layers free.
- Bandwidth control. Constrain only mid-frequencies where symmetry is expected.

8.7 Interaction with common layers and ops

BatchNorm/LayerNorm. Global statistics are usually fine; but if you apply GEP (GAP$\circ \boldsymbol{P}_{+}$) just before logits, ensure training/eval modes are correct to avoid statistic drift.
Strided/“same” convs. Stride and padding break exact equivariance; prefer odd kernel sizes and symmetric padding in parity-constrained blocks.
Nonlinearities. Pointwise activations commute with $\boldsymbol{R}$; pooling with asymmetric windows does not—use symmetric windows around centers.

8.8 Non-abelian or irregular domains

Problem. DFT-based $\boldsymbol{R}$assumes an abelian grid/torus. Graphs/meshes and non-abelian groups need different machinery.
Mitigations.

- Group-specific constructions. Use harmonic bases or irreps for the domain; define inversion via the group antipode.
- Graph automorphisms. If an order-2 automorphism exists, average over $\{\boldsymbol{I}, \boldsymbol{\iota}\}$to build a graph-parity projector.

8.9 Distribution shift and dataset bias

Problem. Training data may be parity-balanced, deployment data not (or vice versa).
Mitigations.

- Track odd-leakage $\boldsymbol{\rho} = \| \boldsymbol{P}\_\boldsymbol{\phi}(\boldsymbol{x}) \| \frac{}{\|\boldsymbol{\phi}(\boldsymbol{x})} \|$ online; rising $\boldsymbol{\rho}$ flags shift.
- Keep DM tools auditable: ability to disable or reduce parity at inference via learned gates.

8.10 Generative dullness ("over-symmetrization")

Problem. Generators trained with strong $\boldsymbol{R}$-penalties may produce overly centered, bland samples.
Mitigations.

- Apply DM penalties to features (U-Net bottleneck) not outputs; keep a small $\boldsymbol{\lambda}$.
- Use fractional $\boldsymbol{F}_{\boldsymbol{\alpha}}^{\boldsymbol{2}}$ curriculum rather than hard parity; enforce only in a subset of steps or bands.

8.11 Compute/engineering caveats

Problem. Two FFTs are cheap, but on tiny inputs overhead can be non-negligible; mixed-precision FFT support varies.
Mitigations.

- Batch and fuse FFT calls; restrict DM ops to a few layers/epochs.
- Verify backend supports your precision; fall back to float32 for FFT while keeping rest in bf16/fp16.

8.12 Quick red-flag checklist

- Accuracy down on asymmetric classes after enabling DM → turn on gating or disable per class.
- Spikes at borders in $\boldsymbol{P}\_\boldsymbol{x}$→ use reflection padding/tapers.
- Odd content completely gone but task needs handedness → move constraint to features / reduce $\boldsymbol{\lambda}$.

- Generations look “too centered” → weaken penalty, apply in bands, or use fractional $\boldsymbol{\alpha}$.

Bottom line. Dual-mirror tools are powerful when parity is real, and informative as audits when it isn’t. Treat them as opt-in structure—selectively, locally, and with diagnostics—rather than a universal assumption.

9. Outlook: Beyond Parity

The “two FFTs + an index flip” mindset scales naturally to a broader metaplectic toolbelt—operators that act unitarily on $\boldsymbol{L^2}$ and correspond to symplectic maps of phase space. Parity $\boldsymbol{R = F^2}$ is only one vertex of this group. Generalizing the dual-mirror philosophy yields structure-preserving augmentations, priors, and diagnostics that remain cheap, differentiable, and algebraically controlled.

9.1 Shears (chirps) as learnable, physics-faithful augmentations

Operators. Time/frequency shears correspond to multiplying by (or convolving with) chirps:

- Time-chirp (quadratic phase): $(C_a f)(x) = e^{i\pi a\|x\|^2} f(x)$.
- Frequency-chirp: $\widehat{C_b f}(\xi) = e^{i\pi b\|\xi\|^2} \hat{f}(\xi)$.
  Together with $F$and scalings, these generate linear canonical transforms (LCTs)—the metaplectic lifts of $SL(2,\mathbb{R})$ (or $Sp(2d,\mathbb{R})$ in dD).

Use cases.

- Data-consistent augmentations: emulate dispersion, defocus, and sweep-rate mismatch (radar/sonar), lens aberrations (optics), or frequency glide (audio) *without* breaking physical constraints.
- Chirp-equivariant layers: tie kernels to commute with a chosen shear; audit violations via $\| UKU^{-1} - K \|$for metaplectic U.

Learnable parameters. Make the shear strengths (a, b)trainable with small priors; anneal or regularize toward task-specific physical ranges.

9.2 Fractional rotations F_α: continuous curricula and diagnostics

Phase-space rotation. F_α is rotation by $\alpha \cdot \left(\frac{\pi}{2}\right)$ in (x, ξ). Parity is the half-turn ($\alpha = 2$); identity is $\alpha = 0$.

Curricula.

- Robustness sweeps: enforce or test $\phi(x) \approx \phi(F_\alpha^2 x)$ for a schedule α_t that moves from easy (near 0) to hard (near 2).
- Stiffness probes: measure how representation distortion grows with α; flat slopes indicate symmetry-aware features.

Practical knobs.

- Band-limited F_α: apply rotations only to mid-frequencies to avoid boundary/aliasing artifacts.
- Stochastic α: sample $\alpha \sim \mathcal{U}(\alpha_{\min}, \alpha_{\max})$ per batch; log a "rotation sensitivity" curve.

9.3 Learned symmetry schedules (operator policies)

Instead of hard-coding a single operator, treat a *mixture* as a policy:

$$\mathcal{U}_\theta = \sum_k w_k(\theta)\, U_k, \quad U_k \in \{I,\ F,\ F_\alpha^2,\ C_a,\ \text{parity } R,\ \text{scalings}\}.$$

Training strategies.

- Bilevel: inner loop trains the model; outer loop tunes the operator weights w_k for robustness/metric targets.
- Band/region gating: learn spatial–spectral masks selecting which operator(s) act where.
- Cost-aware policies: penalize operators by compute or risk of prior mismatch; the schedule learns when structure helps.

9.4 Wigner/LCT views for interpretability

Use Wigner distributions or LCT spectrograms to visualize how features transform under metaplectic elements. Stable models show coherent rotations/shears; unstable ones exhibit tearing or phase scrambling. This is a principled lens for diagnosing invariance beyond simple flips.

9.5 From parity to proxy-invariance (ethics)

Parity offers a sandbox for counterfactual invariance: outcomes shouldn't depend on a transformation deemed irrelevant. Generalize this:

- Orientation proxy. If orientation correlates with a sensitive attribute (e.g., camera mounting biases), use P_+-based audits to detect leakage *without accessing* protected labels:

$$\Delta_{\text{proxy}} = \mathbb{E}[\| g(x) - g(Rx) \|].$$

Large Δ_{proxy} flags orientation-driven outcomes.

- Counterfactual testing. For any sanctioned operator U (parity, mild shear/defocus), enforce $g(x) \approx g(Ux)$ where domain knowledge says outcomes should be invariant. This operationalizes fairness without demographic data—through physics/geometry-based proxies.
- Selective fairness. Apply invariance only in subspaces where labels are provably unchanged; use gates/masks to avoid hiding legitimate cues (e.g., chiral molecules, traffic direction).

9.6 Research directions (actionable)

1. Metaplectic layers as primitives. Drop-in PyTorch modules for F_α, chirps, scalings with stable gradients, mixed precision, and batched N-D support.
2. Operator-aware learning theory. Generalization bounds under group-average projections for continuous families; sample complexity vs operator entropy.
3. Fractional curricula schedulers. Learn α_t (and bands) jointly with the model to optimize robustness curves.
4. Shear-equivariant attention. Parameterize attention kernels in LCT coordinates; tie or penalize to respect chosen shears.
5. Proxy-invariance benchmarks. Public suites where target labels are invariant to known metaplectic transforms; include orientation-bias stress tests and evaluation metrics like Δ_{proxy}.
6. Hybrid audit packs. Wigner-based dashboards that report invariance, stiffness, and odd-leakage across operators.

9.7 Implementation notes (keeping it cheap)

- Operator fusion. Compose metaplectic steps (chirp → FFT → chirp) to reduce FFT count (e.g., Fresnel integrals with two multiplications + one FFT).
- Caching. Precompute chirp masks per resolution; reuse across batches.

- Band-selective ops. Apply in frequency tiles to cut cost and limit artifacts.

Bottom line. Parity is the entry point. The same operator-centric approach extends to rotations, shears, scalings, and their mixtures, giving you a principled way to encode physics, curate augmentations, diagnose representations, and even formalize proxy-invariance—all with light, algebraically controlled modules that play well with modern AI stacks.

10. Conclusion

The dual-mirror perspective reframes a textbook identity—F^2 = reflection—into a practical, operator-centric toolkit. By treating R := F^2 and its projectors P_+ = (I+R)/2, P_- = (I−R)/2 as first-class citizens, we obtain invariants, disentanglers, losses, audits, and curricula that drop into modern stacks with only two FFTs and an index flip. The gains are concrete: better generalization on symmetry-rich data, tighter calibration, more stable reconstructions, and transparent diagnostics (odd-leakage meters, kernel audits) that expose when models violate known structure.

Conceptually, the value comes from algebraic control. Projections like P_+ are norm-nonexpansive, parameter-free, and composable; they shrink hypothesis space exactly where physics or semantics demand it. Practically, the modules are cheap, differentiable, and portable across modalities (images, audio, volumes, time series) and architectures (CNNs, transformers, VAEs, diffusion). When parity is only partially true, local/band-selective and gated variants retain benefits while avoiding bias.

Beyond parity, the same design philosophy extends to the metaplectic family—fractional rotations F_alpha, shears (chirps), and their mixtures—opening a path to learned symmetry schedules and physically faithful augmentations. This suggests a broader program: an operator-centric design language where simple, unitary, algebraically defined transforms become everyday building blocks for robustness, interpretability, and fairness (proxy-invariance without protected attributes).

In short: elevate symmetries from augmentation tricks to operators. Start with parity; add rotations and shears as needed. With minimal code and clear guarantees, you get models that not only perform better where structure matters, but also explain themselves through the invariances they honor.

References

Core Fourier, harmonic analysis, and time–frequency

- Bracewell, R. N. *The Fourier Transform and Its Applications* (3rd ed.). McGraw-Hill, 2000.
- Goodman, J. W. *Introduction to Fourier Optics* (4th ed.). W. H. Freeman, 2017.
- Mallat, S. *A Wavelet Tour of Signal Processing* (3rd ed.). Academic Press, 2008.
- Grochenig, K. *Foundations of Time-Frequency Analysis*. Birkhäuser, 2001.
- Cohen, L. *Time-Frequency Analysis*. Prentice Hall, 1995.
- Folland, G. B. *Fourier Analysis and Its Applications*. American Mathematical Society, 2009.
- Rudin, W. *Fourier Analysis on Groups*. Wiley, 1962.
- Folland, G. B. *A Course in Abstract Harmonic Analysis* (2nd ed.). CRC Press, 2016.
- Poularikas, A. D. (Ed.). *The Transforms and Applications Handbook* (3rd ed.). CRC Press, 2010.
- Papoulis, A. *Signal Analysis*. McGraw-Hill, 1977.

Fractional Fourier / linear canonical transforms / metaplectic

- Ozaktas, H. M., Zalevsky, Z., & Kutay, M. A. *The Fractional Fourier Transform: With Applications in Optics and Signal Processing*. Wiley, 2001.
- de Gosson, M. A. *Symplectic Methods in Harmonic Analysis and in Mathematical Physics*. Birkhäuser, 2011.
- Folland, G. B. *Harmonic Analysis in Phase Space*. Princeton University Press, 1989.
- Hlawatsch, F., & Auger, F. (Eds.). *Time-Frequency Analysis: Concepts and Methods*. Elsevier, 2013.

Group/equivariant learning and invariances in ML

- Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press, 2016.
- Cohen, T., & Welling, M. “Group Equivariant Convolutional Networks.” *ICML*, 2016.
- Kondor, R., & Trivedi, S. “On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups.” *ICML*, 2018.

- Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. MIT Press, 2023.
- Shorten, C., & Khoshgoftaar, T. M. "A Survey on Image Data Augmentation for Deep Learning." *Journal of Big Data* 6, 60 (2019).

Diffusion / score-based generative modeling (for §5.3)

- Ho, J., Jain, A., & Abbeel, P. "Denoising Diffusion Probabilistic Models." *NeurIPS*, 2020.
- Song, Y., & Ermon, S. "Generative Modeling by Estimating Gradients of the Data Distribution." *NeurIPS*, 2019.
- Song, Y., et al. "Score-Based Generative Modeling through Stochastic Differential Equations." *ICLR*, 2021.

Calibration, robustness, OOD diagnostics

- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. "On Calibration of Modern Neural Networks." *ICML*, 2017.
- Hendrycks, D., & Gimpel, K. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." *ICLR*, 2017.

SAR / ISAR / radar (symmetry-rich domains)

- Cumming, I. G., & Wong, F. H. *Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation*. Artech House, 2005.
- Curlander, J. C., & McDonough, R. N. *Synthetic Aperture Radar: Systems and Signal Processing*. Wiley, 1991.
- (Venue-specific add: papers on 180° invariance/centrosymmetry in SAR target recognition and reconstruction.)

Crystallography / diffraction / phase retrieval

- Giacovazzo, C. (Ed.). *Fundamentals of Crystallography* (3rd ed.). Oxford University Press, 2011.
- Patterson, A. L. "A Direct Method for the Determination of the Components of Interatomic Distances in Crystals." *Zeitschrift für Kristallographie* 90, 517–542 (1935).
- Gerchberg, R. W., & Saxton, W. O. "A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures." *Optik* 35, 237–246 (1972).

- Fienup, J. R. “Phase Retrieval Algorithms: A Comparison.” *Applied Optics* 21(15), 2758–2769 (1982).
- Marchesini, S. “A Unified Evaluation of Iterative Projection Algorithms for Phase Retrieval.” *Review of Scientific Instruments* 78, 011301 (2007).
- (Add domain papers on centrosymmetry constraints in CDI/ptychography for your venue.)

Medical imaging and acoustics

- Hansen, P. C., Nagy, J. G., & O’Leary, D. P. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, 2006.
- Natterer, F. *The Mathematics of Computerized Tomography*. SIAM, 2001.
- Kuttruff, H. *Room Acoustics* (6th ed.). CRC Press, 2016.
- (Add modality-specific works where even/odd decomposition or parity priors are standard.)

Fairness / proxy-invariance (for §9)

- Kusner, M. J., Loftus, J., Russell, C., & Silva, R. “Counterfactual Fairness.” *NeurIPS*, 2017.
- Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. “Distributionally Robust Neural Networks.” *ICLR*, 2020.
- (Use these as conceptual anchors when framing orientation-proxy invariance without protected attributes.)

Note. For submission, replace parenthetical prompts with concrete, venue-appropriate citations (e.g., specific SAR parity studies, centrosymmetric CDI results, medical phantom parity priors). Where possible, include DOIs and arXiv IDs to ease reproducibility.

Appendix A: Minimal Spec

A.0 Notation and assumptions

- Domain: d-dimensional periodic grid of shape D1×…×Dd.
- Tensors: x ∈ R^{N×C×D1×…×Dd} (batch N, channels C). Complex tensors allowed.
- FFTs are unitary (norm='ortho') and applied over the d spatial dims only.
- reverse_indices(·) performs circular index reversal k ↦ (−k) mod D along each spatial dim.

A.1 Core operators

Dual mirror (parity)

- Continuous: R f(x) = f(−x).
- Discrete (DFT grid):
  R(x) = reverse_indices(IFFT(FFT(x))) // unitary FFTs

Projectors

- P_plus(x) = 0.5 * (x + R(x)) // even part
- P_minus(x) = 0.5 * (x − R(x)) // odd part
- Properties: P_plus^2 = P_plus, P_minus^2 = P_minus, P_plus P_minus = 0, R = P_plus − P_minus.

Dual-mirror consistency loss (feature space)

- L_DM(ϕ; x) = || ϕ(x) − ϕ(R(x)) ||_2^2

Kernel audit (linear operator K)

- eps_plus(K) = || R K R − K || / ||K|| // deviation from even equivariance
- eps_minus(K) = || R K R + K || / ||K|| // deviation from odd equivariance

Norms are any operator or Frobenius norms consistent across comparisons.

A.2 Reference implementations (framework-agnostic pseudocode)

def dual_mirror(x, spatial_dims):

```
    X  = fftn(x, dim=spatial_dims, norm='ortho')
    x2 = ifftn(X, dim=spatial_dims, norm='ortho')
    return flip_circular(x2, dims=spatial_dims)   # reverse_indices

def P_plus(x, spatial_dims):
    return 0.5 * (x + dual_mirror(x, spatial_dims))

def P_minus(x, spatial_dims):
    return 0.5 * (x - dual_mirror(x, spatial_dims))

def L_DM(phi, x, spatial_dims):
    return l2sq(phi(x) - phi(dual_mirror(x, spatial_dims)))

def kernel_audit_even(K_apply, x, spatial_dims):
    # K_apply: function applying K to a tensor x
    num = l2sq(dual_mirror(K_apply(dual_mirror(x, spatial_dims)), spatial_dims) - K_apply(x))
    den = l2sq(K_apply(x)) + 1e-12
    return (num / den)**0.5

def kernel_audit_odd(K_apply, x, spatial_dims):
    num = l2sq(dual_mirror(K_apply(dual_mirror(x, spatial_dims)), spatial_dims) + K_apply(x))
    den = l2sq(K_apply(x)) + 1e-12
    return (num / den)**0.5
```

Notes

- flip_circular must reverse each spatial axis with wraparound (index k→−k mod D).
- For real inputs, take .real after dual_mirror if your pipeline expects real tensors.

- Add a tiny epsilon in denominators to avoid division by zero.

A.3 PyTorch-style snippets (drop-in)

```
import torch


def dual_mirror(x, spatial_dims=(-2, -1)):
   X  = torch.fft.fftn(x, dim=spatial_dims, norm='ortho')
   x2 = torch.fft.ifftn(X, dim=spatial_dims, norm='ortho')
   return x2.flip(dims=spatial_dims).real if x.is_floating_point() else x2.flip(dims=spatial_dims)


def P_plus(x, spatial_dims=(-2, -1)):
   return 0.5 * (x + dual_mirror(x, spatial_dims))


def P_minus(x, spatial_dims=(-2, -1)):
   return 0.5 * (x - dual_mirror(x, spatial_dims))


def dm_loss(phi, x, spatial_dims=(-2, -1)):
   return torch.mean((phi(x) - phi(dual_mirror(x, spatial_dims)))**2)
```

A.4 Fractional path (optional extension)

Fractional Fourier transform F_alpha implements a phase-space rotation by alpha*(pi/2). A fractional dual mirror uses F_alpha^2:

- R_alpha(x) = F_alpha^{-1}(F_alpha(x)) with appropriate index reversal
- L_DM_alpha = || ϕ(x) − ϕ(R_alpha(x)) ||_2^2

Practical construction uses chirp–FFT–chirp factorizations (linear canonical transforms) with two multiplications + one FFT; cache chirp masks per resolution.

A.5 Local and band-limited variants

Local (patch) projection

1. Extract overlapping windows W_i x.
2. y_i = P_plus(W_i x).
3. Reconstruct with smooth overlap-add: x_even = SOA({y_i}).

Band-limited projection

- In Fourier domain, apply P_plus only on a mask B(ξ):
  x_even = F^{-1}[B(ξ) * 0.5(X(ξ) + X(−ξ)) + (1−B(ξ)) * X(ξ)].

Use these when parity holds only in certain regions/bands.

A.6 Boundary and padding policy

- FFT assumes periodic wraparound. For nonperiodic data, apply reflection padding or a Tukey window before dual_mirror.
- Document pad widths or taper parameters; parity artifacts otherwise concentrate at borders (large P_minus near edges).

A.7 Complexity and precision

- Each call to R or P_plus/P_minus adds two FFTs over spatial dims: O(N log N).
- Overhead is typically negligible compared to deep backbones; still, measure ms/batch.
- Prefer float32/float64 for phase-sensitive tasks; bf16/fp16 are generally fine for classification/regression.

A.8 Minimal unit tests (sanity)

1. Idempotence
    - assert_close(P_plus(P_plus(x)), P_plus(x))
    - assert_close(P_minus(P_minus(x)), P_minus(x))
2. Orthogonality

- assert_close(inner(P_plus(x), P_minus(x)), 0, atol≈1e−6)

3. Decomposition
    - assert_close(P_plus(x) + P_minus(x), x)
4. Involution
    - assert_close(dual_mirror(dual_mirror(x)), x) // up to numerical tolerance
5. Energy non-expansive
    - assert(||P_plus(x)||_2 <= ||x||_2 + 1e−6), same for P_minus.

A.9 GEP head (one-liner)

- GEP(x) = GAP(P_plus(x))
  Implementation: even_feat = 0.5*(feat + dual_mirror(feat)); out = even_feat.mean(dim=spatial_dims)

A.10 Training hooks (short recipes)

- Consistency: add λ * dm_loss(phi, x) at a mid layer; λ ∈ [1e−2, 1e−1].
- Kernel tying (soft): add λ_tie * || R K R − K ||^2 (even) or λ_tie * || R K R + K ||^2 (odd) per layer.
- Audit: log rho = ||P_minus(x)||/||x|| and invariance score Inv = E||ϕ(x)−ϕ(Rx)||/E||ϕ(x)||.

A.11 Interface summary (ready to paste)

- dual_mirror(x, spatial_dims) → tensor like x
- P_plus(x, spatial_dims) → tensor like x
- P_minus(x, spatial_dims) → tensor like x
- dm_loss(phi, x, spatial_dims) → scalar loss
- kernel_audit_even(K_apply, x, spatial_dims) → scalar in [0,∞)
- kernel_audit_odd(K_apply, x, spatial_dims) → scalar in [0,∞)

These building blocks are sufficient to reproduce all experiments described in §5 with only two FFTs and an index flip per use.

Appendix B: Evaluation Checklist

B.1 Symmetry sanity checks (before any training)

- Label invariance test: for each class c, verify y(x) == y(Rx); if not, record permutation/sign map Pi_c or mark class as non-invariant.
- Invariance stratification: split dataset into invariant vs non-invariant subsets; report counts and example visuals.
- Invariance score (raw data): Inv_data = E[||x − Rx|| / ||x||] per class; flag large values where parity prior may be false.
- Centering/alignment: confirm symmetry center at origin/grid center; log any recentering transforms.

B.2 Preprocessing and boundaries

- FFT normalization: use unitary FFTs (norm='ortho'); state library/version.
- Padding policy: reflection padding / taper (e.g., Tukey a=0.25); document exact parameters and where applied.
- Windowed parity (if used): window size, stride, overlap-add scheme; provide code.
- Precision: fp16/bf16 vs fp32/fp64 choices; justify for phase-sensitive tasks.

B.3 Implemented components (on/off toggles)

- P_+ projection: location(s) in the model (input, mid-layer, head).
- L_DM (dual-mirror loss): layer(s) applied, weight lambda, schedule/anneal.
- Kernel tying: layers constrained (even/odd), hard- vs soft-tying (lambda_tie).
- GEP head: GAP(P_+(feat)) before logits (yes/no).
- Fractional path: F_alpha^2 usage; alpha schedule or sampling distribution.
- Gating: learned gate g(x) in [0,1] for class-/region-conditional application (yes/no).

B.4 Training protocol

- Seeds: at least 5 fixed seeds; report mean ± std.
- Splits: stratified train/val/test; publish indices.
- Optimizer: AdamW/SGD params (lr, wd, betas, schedule).

- Augmentation: standard stack, plus parity aug (x,Rx) rates if used.
- Early stopping / selection: criterion (val accuracy, ECE, composite).

B.5 Metrics (primary and secondary)

- Accuracy / mAP on full and invariant-only subsets.
- Calibration: ECE (bins specified), Brier score; reliability plots released.
- OOD under R: AUROC distinguishing x vs Rx; target near 0.5 when parity holds.
- Robustness: accuracy under parity-consistent corruptions (blur, speckle, low-pass); corruption strengths specified.
- Invariance score (features): Inv_feat = E[||phi(x) − phi(Rx)|| / ||phi(x)||] per layer; plot vs depth.
- Odd-leakage: rho = ||P_- x|| / ||x|| (or in feature space); histograms per class/domain.
- Generative (if applicable): FID/KID; human “balance” preference with n, CI, and kappa.

B.6 Kernel and operator audits

- Kernel parity errors: eps_plus = ||RKR − K|| / ||K||, eps_minus = ||RKR + K|| / ||K|| per block; report means and top-5 offenders.
- Ablation-by-layer: turn parity tying on/off per block; show metric deltas.
- Operator stiffness: sensitivity curve for F_alpha^2 sweep (alpha in [0,2]) on features and outputs.

B.7 Ablations (minimal set)

- P_+ only vs L_DM only vs kernel tying only vs GEP only vs full combo.
- Loss weight sweep: lambda in {0.01, 0.05, 0.1}; report Pareto curves (accuracy vs ECE).
- Padding alternatives: reflection vs circular vs taper; report boundary artifact metrics (P_- energy near edges).
- Placement: apply constraints at early vs middle vs late blocks; compare.

B.8 Compute, memory, and engineering

- Overhead: median ms/batch with and without DM ops (forward and train); input sizes specified.
- FFT cost breakdown: fraction of step time spent in FFTs.

- Memory: peak VRAM/host RAM deltas.
- Hardware/env: GPU/CPU models, library versions, mixed-precision settings.

B.9 Reproducibility package

- Code release: dual_mirror/P_plus/P_minus operators; DM loss; kernel-tying wrappers; metrics.
- Configs: YAML/JSON for all experiments; seed lists; dataset download/prepare scripts.
- Exact padding/window params and any centering transforms.
- Result artifacts: CSVs of metrics per epoch/seed; plots; sample visuals of P_+ vs P_- decompositions.
- Unit tests: idempotence, orthogonality, decomposition, energy non-expansive.

B.10 Risk and fairness notes

- False-prior audit: per-class performance delta when turning DM on; disable or gate for harmed classes.
- Proxy-invariance check: delta_proxy = E[||g(x) − g(Rx)||]; flag orientation leakage without protected attributes.
- Disclosure statement: where parity is assumed, where it is optional/gated, and known failure cases.

B.11 Acceptance criteria (quick pass/fail)

- Documented padding and FFT normalization.
- Reported overhead (ms/batch) and per-component ablation.
- Released toggles for class-conditional parity (config or flags).
- Posted seeds, splits, and code sufficient to reproduce ±1% main metrics.