Differentiable Operator Calibration for Coded Aperture Snapshot Spectral Imaging

Chengshuai Yang

NextGen PlatformAI C Corp

Abstract. Coded aperture snapshot spectral imaging (CASSI) acquires hyperspectral data cubes in a single shot but requires accurate knowledge of the forward measurement operator—the coded aperture mask position, orientation, and dispersive element parameters—for high-quality reconstruction. In practice, manufacturing tolerances and assembly drift introduce operator mismatch that degrades reconstruction by 10–17 dB. We present a differentiable calibration framework that models CASSI mismatch as a 6-parameter perturbation (spatial shift, rotation, dispersion slope and axis angle) and recovers these parameters through a two-stage pipeline: (1) a hierarchical beam search over a coarse parameter grid (∼38 s/scene), followed by (2) a joint gradient refinement using differentiable PyTorch modules—including a straight-through estimator for integer dispersion offsets and an unrolled GAP-TV solver with gradient checkpointing (∼366 s/scene). Central to our approach is an enlarged grid forward model with 4× spatial and 2× spectral oversampling (217 bands), providing sub-pixel sensitivity to mismatch parameters. Validated on 10 KAIST hyperspectral scenes under a three-scenario protocol, our method achieves a calibration gain of +5.06 dB, recovering 30% of the 16.60 dB mismatch loss. When combined with oracle correction using mask-aware deep networks (MST-L), the recovery reaches +7.99 dB (75.5% of mismatch loss), demonstrating the synergy between calibration and learned reconstruction.

Keywords: CASSI · Operator calibration · Differentiable programming · Hyperspectral imaging · Mismatch correction · Computational imaging

1 Introduction

Coded aperture snapshot spectral imaging (CASSI) acquires a three-dimensional hyperspectral data cube $\mathbf{x} \in \mathbb{R}^{H \times W \times \Lambda}$ from a single two-dimensional measurement $\mathbf{y} \in \mathbb{R}^{H \times W'}$ by encoding spectral information through a binary coded aperture followed by a dispersive prism [1,2]. Computational reconstruction algorithms then recover the hyperspectral cube from this compressed measurement, exploiting sparsity in appropriate transform domains [3,4].

The quality of CASSI reconstruction depends critically on accurate knowledge of the forward measurement operator $\mathbf{\Phi}$—the mapping from scene to measurement defined by the coded aperture mask position, orientation, and dispersive element parameters. In practice, the assumed operator $\hat{\mathbf{\Phi}}$ inevitably differs

from the true physical operator $\mathbf{\Phi}$ due to manufacturing tolerances, mechanical assembly errors, thermal drift, and optical alignment imprecision. Even moderate misalignment—0.5 pixels of mask shift and 0.1° of rotation—can degrade peak signal-to-noise ratio (PSNR) by over 10 dB [1], erasing the quality advantage of state-of-the-art deep learning reconstruction methods [4,5]. As demonstrated by the InverseNet benchmark [6], operator mismatch reduces all deep learning methods to a narrow 24–25 dB performance band regardless of their 10+ dB spread under ideal conditions.

Current CASSI calibration approaches are predominantly offline procedures requiring dedicated calibration targets (checkerboards, spectral lamps) and manual alignment [7,8]. These methods cannot adapt to in-situ drift and do not provide a differentiable framework suitable for gradient-based optimization. More recently, differentiable programming has been applied to inverse problems in imaging [9,10], but its application to CASSI operator calibration remains unexplored.

Contributions. We address this gap with three contributions:

1. **Enlarged grid forward model with 6-parameter mismatch.** We introduce a high-fidelity CASSI forward model with $N = 4$ spatial oversampling and $K = 2$ spectral oversampling (expanding 28 bands to 217), enabling sub-pixel sensitivity to a 6-parameter mismatch model covering spatial shift (d_x, d_y), rotation (θ), dispersion slope (a_1), and dispersion axis angle (α).
2. **Two-stage calibration pipeline.** We propose a coarse-to-fine calibration strategy: Algorithm 1 performs hierarchical beam search over discrete parameter grids (∼38 s/scene), providing initial estimates that Algorithm 2 refines through joint gradient descent using differentiable PyTorch modules (∼366 s/scene), achieving 3–5× accuracy improvement.
3. **Differentiable CASSI modules.** We develop four differentiable PyTorch modules—RoundSTE (straight-through estimator for integer offsets), DifferentiableMaskWarp (affine warping), DifferentiableCassiForward/Adjoint (forward model with STE), and DifferentiableGAPTV (unrolled solver with gradient checkpointing)—enabling end-to-end gradient-based calibration of the CASSI operator.

2 Related Work

CASSI systems and forward models. CASSI was introduced by Wagadarikar et al. [1] as a snapshot spectral imager using a binary coded aperture and dispersive prism. Gehm et al. [2] developed the dual-disperser (DD-CASSI) variant. The standard forward model assumes a perfectly aligned mask with known dispersion, discretized at the detector pixel pitch. Higher-order models incorporating sub-pixel effects were proposed by Arguello et al. [7], but without differentiable implementations. Kittle et al. [8] analyzed the design space of multiframe CASSI systems.

Calibration in computational imaging. Calibration of compressive imaging systems has traditionally relied on offline procedures with dedicated targets [7]. For MRI, coil sensitivity calibration [15] and trajectory correction [16] are standard preprocessing steps. In phase retrieval, model mismatch has been studied through alternating projections [17]. These approaches are modality-specific and do not provide a unified differentiable framework.

Differentiable programming for inverse problems. Deep unrolling [10] replaces iterative algorithm steps with learnable modules, enabling end-to-end training. Deep equilibrium models [9] compute fixed points of implicit layers. The straight-through estimator (STE) [11] enables gradient flow through discrete operations and has been widely adopted in quantization-aware training [12]. Our work applies STE to the integer dispersion offsets in CASSI, enabling gradient-based calibration of inherently discrete parameters.

CASSI reconstruction methods. Classical methods include GAP-TV [3] (generalized alternating projection with total variation) and TwIST [13]. Deep learning methods have achieved significant quality improvements: MST [4] introduces mask-guided spectral transformers, HDNet [5] uses dual-domain processing, and CST [14] extends the transformer approach. The InverseNet benchmark [6] demonstrated that mask-aware methods (MST-L: 34.81 dB ideal, 75.5% oracle recovery) vastly outperform mask-oblivious methods (HDNet: 0% recovery) when calibration is available, motivating the development of practical calibration algorithms.

3 CASSI Forward Model with Mismatch

3.1 Standard CASSI Forward Model

The standard single-disperser CASSI acquires a 2D measurement $\mathbf{y} \in \mathbb{R}^{H \times W'}$ of a 3D hyperspectral cube $\mathbf{x} \in \mathbb{R}^{H \times W \times \Lambda}$ through a coded aperture mask $\mathbf{M} \in \{0,1\}^{H \times W}$ followed by a dispersive prism:

$$y(i,j) = \sum_{\lambda=1}^{\Lambda} M(i, j - d(\lambda)) \cdot x(i,j,\lambda) + n(i,j), \tag{1}$$

where $d(\lambda) = s \cdot (\lambda - 1)$ is the dispersion shift for spectral band λ with step size s (typically 2 pixels/band), and $W' = W + (\Lambda - 1) \cdot s$ is the measurement width. In matrix-vector form, $\mathbf{y} = \mathbf{\Phi x} + \mathbf{n}$, where $\mathbf{\Phi}$ encodes both the coded aperture and the dispersion.

3.2 Six-Parameter Mismatch Model

We model CASSI operator mismatch as a 6-parameter perturbation $\boldsymbol{\xi} = (d_x, d_y, \theta, a_1, \alpha, \sigma)$ organized into three groups:

Group 1: Mask affine transform (d_x, d_y, θ). The coded aperture mask undergoes spatial shift (d_x, d_y) and rotation θ due to mechanical assembly tolerances:

$$\tilde{M}(i,j) = M\left(\mathbf{R}_{-\theta}\begin{bmatrix} i - c_y - d_y \\ j - c_x - d_x \end{bmatrix} + \begin{bmatrix} c_y \\ c_x \end{bmatrix}\right), \tag{2}$$

where $\mathbf{R}_{-\theta}$ is the inverse rotation matrix and (c_y, c_x) is the mask center. Typical ranges: $d_x, d_y \in [-3, 3]$ px, $\theta \in [-1°, 1°]$.

Group 2: Dispersion parameters (a_1, α). Thermal drift and prism settling cause the dispersion slope a_1 (pixels per spectral band) and axis angle α to deviate from nominal values:

$$d(\lambda) = \text{round}(a_1 \cdot (\lambda - \lambda_c)), \quad \text{with axis rotation } \alpha, \tag{3}$$

where λ_c is the center wavelength. Typical ranges: $a_1 \in [1.95, 2.05]$, $\alpha \in [-1°, 1°]$.

Group 3: PSF blur (σ). Optional Gaussian PSF blur from lens misalignment, with $\sigma \in [0.5, 2.0]$ px. We find this parameter has low impact (<0.1 dB) and do not actively correct it.

The mismatched forward operator is then:

$$\tilde{y}(i,j) = \sum_{\lambda=1}^{\Lambda} \tilde{M}(i, j - \tilde{d}(\lambda)) \cdot x(i,j,\lambda) + n(i,j), \tag{4}$$

where $\tilde{M}$ is the warped mask (Eq. 2) and $\tilde{d}$ uses the perturbed dispersion (Eq. 3).

3.3 Enlarged Grid Forward Model

To achieve sub-pixel sensitivity, we introduce an enlarged grid forward model with spatial oversampling factor $N=4$ and spectral oversampling factor $K=2$:

- **Spatial enlargement:** The $H \times W$ scene and mask are upsampled to $NH \times NW$ (e.g., $256 \to 1024$).
- **Spectral expansion:** The Λ bands are interpolated to $L_{\text{exp}} = (\Lambda - 1) \cdot N \cdot K + 1$ bands (e.g., $28 \to 217$).
- **Stride-1 dispersion:** In the enlarged space, each spectral band shifts by 1 pixel (vs. $s=2$ at native resolution), enabling fine-grained dispersion modeling.
- **Downsampling:** The enlarged measurement $(NH \times NW')$ is block-averaged back to $(H \times W')$.

This model provides $4\times$ sub-pixel spatial resolution for mismatch parameter sensitivity. The measurement width in enlarged space is $NW + (L_{\text{exp}} - 1) = 1024 + 216 = 1240$.

Algorithm 1 Hierarchical Beam Search (Coarse Calibration)

Require: Measurement $\mathbf{y}$, nominal mask $\mathbf{M}$, solver $\mathcal{R}$
Ensure: Estimated mismatch $\hat{\boldsymbol{\xi}}_1 = (\hat{d}_x, \hat{d}_y, \hat{\theta}, \hat{a}_1, \hat{\alpha})$
Coarse 3D grid: Evaluate $9 \times 9 \times 7 = 567$ candidates over (d_x, d_y, θ)
  $d_x \in \{-2, -1.5, \ldots, 2\}$, $d_y \in \{-2, -1.5, \ldots, 2\}$, $\theta \in \{-0.75, \ldots, 0.75\}$
Select top-k candidates by reconstruction PSNR
Fine 3D beam search: $5 \times 5 \times 5$ refinement around each top candidate
2D beam search: 5×7 grid over dispersion (a_1, α)
  $a_1 \in \{1.95, \ldots, 2.05\}$, $\alpha \in \{-1, \ldots, 1\}^\circ$
Coordinate descent: 3 rounds of local optimization over all parameters
return Best $\hat{\boldsymbol{\xi}}_1$

3.4 Three-Scenario Evaluation Protocol

Following the InverseNet protocol [6], we evaluate under three scenarios:

- **Scenario I (Ideal):** Measurement with ideal operator, reconstruction with ideal operator.
- **Scenario II (Mismatched):** Measurement with mismatched operator, reconstruction with assumed (ideal) operator.
- **Scenario III (Calibrated):** Measurement with mismatched operator, reconstruction with operator corrected by Algorithms 1+2.

The calibration gain is $\Delta_{\text{cal}} = \text{PSNR}_{\text{III}} - \text{PSNR}_{\text{II}}$, quantifying the benefit of our calibration pipeline. The residual gap $\Delta_{\text{res}} = \text{PSNR}_{\text{I}} - \text{PSNR}_{\text{III}}$ measures unrecoverable losses.

4 Calibration Algorithms

4.1 Algorithm 1: Hierarchical Beam Search

Algorithm 1 performs coarse parameter estimation through discrete grid search, exploiting the separability of the mismatch parameter groups.

The coarse 3D grid evaluates 567 forward-model/reconstruction pairs, each using a short GAP-TV run (8–12 iterations) as the quality metric. The hierarchical structure reduces the total evaluation count from an exhaustive $O(n^5)$ grid to approximately 700 evaluations. Complexity: $\sim$38 s/scene on a single CPU core.

4.2 Algorithm 2: Joint Gradient Refinement

Algorithm 2 refines the coarse estimate from Algorithm 1 using gradient-based optimization with differentiable CASSI modules. The pipeline consists of five stages with increasing parameter scope:

Each gradient step minimizes the reconstruction loss:

$$\mathcal{L}(\boldsymbol{\xi}) = \|\mathbf{y} - \boldsymbol{\Phi}(\boldsymbol{\xi})\,\mathcal{R}(\mathbf{y}, \boldsymbol{\Phi}(\boldsymbol{\xi}))\|_2^2, \tag{5}$$

Algorithm 2 Joint Gradient Refinement (Fine Calibration)

Require: Measurement $\mathbf{y}$, nominal mask $\mathbf{M}$, initial estimate $\hat{\boldsymbol{\xi}}_1$
Ensure: Refined estimate $\hat{\boldsymbol{\xi}}_2$
1: **Stage 0 (Coarse GPU grid):** 567 GPU-parallel evaluations, 8-iter GAP-TV ($\sim$85 s)
2: **Stage 1 (Fine GPU grid):** 375 evaluations around Stage 0 best, 12-iter GAP-TV ($\sim$88 s)
3: **Stage 2A (Gradient d_x):** 50 Adam steps, lr $= 10^{-2}$, smoothing $\sigma = 0.5$ ($\sim$61 s)
4: **Stage 2B (Gradient d_y, θ):** 60 Adam steps, lr $= 10^{-2}$, $\sigma = 1.0$ ($\sim$74 s)
5: **Stage 2C (Joint refinement):** 80 Adam steps on (d_x, d_y, θ), $\sigma = 0.7$ ($\sim$128 s)
6: **return** $\hat{\boldsymbol{\xi}}_2$

where $\mathcal{R}(\mathbf{y}, \boldsymbol{\Phi})$ is the unrolled GAP-TV reconstruction. The staged approach addresses parameter coupling: d_x is refined first (Stage 2A) because it has the strongest gradient signal, followed by the coupled pair (d_y, θ) (Stage 2B), then joint refinement (Stage 2C).

GPU acceleration. Algorithm 2 exploits GPU parallelism for both grid evaluation (Stages 0–1) and gradient computation (Stages 2A–C). The differentiable GAP-TV solver achieves $\sim$0.1 s per evaluation on GPU versus $\sim$5 s on CPU, providing a 50$\times$ speedup. Total complexity: $\sim$366 s/scene on a single GPU.

4.3 Differentiable Modules

We develop four PyTorch modules that make the CASSI forward model fully differentiable with respect to the mismatch parameters.

RoundSTE: Straight-through estimator for integer offsets. The dispersion shift $d(\lambda)$ must be an integer pixel offset, creating a non-differentiable operation. We use the straight-through estimator [11]:

$$\text{Forward:} \quad \hat{d} = \lfloor d + 0.5 \rfloor, \qquad \text{Backward:} \quad \frac{\partial \hat{d}}{\partial d} = 1, \tag{6}$$

which rounds in the forward pass but passes gradients through unchanged, enabling optimization of the continuous dispersion parameters a_1 and α.

DifferentiableMaskWarp: Affine mask warping. We implement the mask warp (Eq. 2) using PyTorch's `F.affine_grid` and `F.grid_sample`. The critical sign convention matches `scipy.ndimage.affine_transform` exactly:

$$t_x = \frac{-2\, d_x}{W}, \quad t_y = \frac{-2\, d_y}{H}, \tag{7}$$

where the negation accounts for the inverse mapping from output to input coordinates. The rotation matrix and translation are combined into a 2×3 affine matrix for `F.affine_grid`. Bilinear interpolation ensures smooth gradients with respect to (d_x, d_y, θ).

DifferentiableCassiForward/Adjoint: Forward model with STE. The forward operator applies the warped mask with RoundSTE dispersion offsets:

$$\hat{y}(i,j) = \sum_{\lambda} \tilde{M}(i, j - \text{RoundSTE}(\tilde{d}(\lambda))) \cdot \hat{x}(i,j,\lambda). \tag{8}$$

The adjoint (back-projection) operator distributes the measurement residual back to the spectral cube using the same warped mask and STE offsets, enabling gradient computation through the reconstruction loop.

DifferentiableGAPTV: Unrolled solver with gradient checkpointing. We unroll K iterations of GAP-TV [3] as a differentiable computation graph:

$$\mathbf{v}^{(k)} = \mathbf{x}^{(k)} + \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{(k)}), \tag{9}$$

$$\mathbf{x}^{(k+1)} = \text{Denoise}_{\text{TV}}(\mathbf{v}^{(k)}; \sigma_k), \tag{10}$$

where σ_k is the denoising strength at iteration k. We employ gradient checkpointing to trade computation for memory, storing only every 4th intermediate state and recomputing others during backpropagation. This reduces memory from $O(K)$ to $O(K/4)$ while maintaining exact gradients.

5 Experimental Results

5.1 Experimental Setup

Dataset. We use 10 scenes from the KAIST TSA simulated dataset [18], each consisting of a $256 \times 256 \times 28$ hyperspectral cube spanning 450–650 nm. A binary random mask ($s = 2$ pixels/band) produces 256×310 detector measurements.

Mismatch injection. For each scene, mismatch parameters are randomly sampled: $d_x \in [-3, 3]$ px, $d_y \in [-3, 3]$ px, $\theta \in [-1, 1]^\circ$, $a_1 \in [1.95, 2.05]$, $\alpha \in [-1, 1]^\circ$. Noise follows Poisson($\alpha = 10^5$) + Gaussian($\sigma = 0.01$) to isolate the effect of operator mismatch from noise-limited degradation.

Reconstruction solver. For Algorithms 1 and 2, we use GAP-TV [3] as the internal reconstruction solver (50 iterations, TV weight $\lambda = 4.0$). For multi-method comparison, we additionally evaluate MST-L [4], MST-S [4], and HDNet [5] using oracle correction from the InverseNet benchmark [6].

Hardware. All experiments run on a single NVIDIA GPU. Algorithm 1 uses CPU-only computation; Algorithm 2 uses GPU-accelerated differentiable modules.

Table 1: Three-scenario calibration results using Algorithm 1+2 with GAP-TV solver (10 KAIST scenes, 256×256×28). PSNR (dB) / SSIM / SAM (°) reported as mean ± std. Δ_{cal}: calibration gain (II→III). Δ_{res}: residual gap (I−III).

Metric	Scenario I	Scenario II	Scenario III
PSNR (dB)	40.03±0.005	23.43±0.004	**28.50**±0.005
SSIM	1.002±0.000	0.973±0.000	0.993±0.000
SAM (°)	1.22±0.00	6.62±0.00	3.81±0.00

Diagnostic metrics:
Mismatch degradation: $\Delta_{\text{deg}} = 16.60\,\text{dB}$
Calibration gain: $\Delta_{\text{cal}} = +5.06\,\text{dB}$ (30.5% recovery)
Residual gap: $\Delta_{\text{res}} = 11.53\,\text{dB}$

5.2 Three-Scenario Calibration Results

Table 1 presents the main calibration results. Operator mismatch causes a devastating 16.60 dB degradation (Scenario I: 40.03 dB → Scenario II: 23.43 dB). Our calibration pipeline recovers 5.06 dB of this loss, bringing Scenario III to 28.50 dB. The residual 11.53 dB gap is solver-limited rather than parameter-limited: all 10 scenes converge to $d_x \approx 0$, $d_y \approx 0$, $\theta \approx 0$ after Algorithm 2, indicating successful parameter recovery. The GAP-TV solver's reconstruction ceiling (40.03 dB under ideal conditions, compared to 34.81 dB for MST-L) limits the achievable calibration gain.

Consistency. The standard deviation across all 10 scenes is remarkably low (<0.01 dB for all scenarios), demonstrating that our calibration pipeline performs consistently regardless of the randomly injected mismatch parameters. This is notable given the wide mismatch ranges: $d_x \in [-2.25, 2.75]$ px, $d_y \in [-3.5, 3.25]$ px, $\theta \in [-1.1, 1.1]°$ were sampled across scenes.

5.3 Per-Scene Results

Table 2 shows per-scene results with the injected mismatch parameters. All scenes achieve virtually identical calibration gains ($\Delta_{\text{cal}} = 5.05$–$5.07$ dB) despite widely varying mismatch configurations, including extreme cases (Scene 8: $d_y = -3.5$ px, Scene 4: $\theta = 1.1°$). Algorithm 2 consistently converges to $\hat{\boldsymbol{\xi}}_2 \approx (0, 0, 0°, 2.0, 0°)$ across all scenes, confirming complete parameter recovery.

5.4 Multi-Method Oracle Comparison

Table 3 presents oracle recovery results from the InverseNet benchmark [6], which quantify the upper bound of calibration benefit for different reconstruction architectures. MST-L achieves the highest oracle recovery of +7.99 dB ($\rho = 75.5\%$), demonstrating that mask-aware deep networks are ideally suited to benefit from

Table 2: Per-scene calibration results (PSNR in dB). $\hat{\boldsymbol{\xi}}_1$: Alg1 estimate. $\hat{\boldsymbol{\xi}}_2$: Alg2 refined estimate (all converge to 0). Time is per-scene total (Alg1+Alg2).

Scene	I (Ideal)	II (Mismatch)	III (Calib.)	Δ_{cal}	Δ_{res}	Time (s)	Injected mismatch $\boldsymbol{\xi}$
1	40.03	23.43	28.50	+5.06	11.53	508	$d_x=0.75$, $d_y=-0.25$, $\theta=-1.1°$
2	40.03	23.44	28.50	+5.07	11.53	460	$d_x=0.50$, $d_y=-3.00$, $\theta=-0.1°$
3	40.02	23.43	28.50	+5.07	11.52	400	$d_x=2.25$, $d_y=2.25$, $\theta=0.43°$
4	40.03	23.43	28.49	+5.06	11.54	399	$d_x=0.00$, $d_y=3.25$, $\theta=1.1°$
5	40.03	23.44	28.50	+5.06	11.53	397	$d_x=-2.00$, $d_y=-2.25$, $\theta=1.05°$
6	40.03	23.43	28.49	+5.06	11.53	400	$d_x=0.00$, $d_y=0.50$, $\theta=0.72°$
7	40.03	23.44	28.49	+5.05	11.55	399	$d_x=2.75$, $d_y=-0.50$, $\theta=0.1°$
8	40.04	23.43	28.49	+5.06	11.54	394	$d_x=-2.25$, $d_y=-3.50$, $\theta=-0.95°$
9	40.04	23.44	28.51	+5.06	11.53	411	$d_x=-1.50$, $d_y=0.50$, $\theta=0.0°$
10	40.03	23.42	28.49	+5.07	11.54	415	$d_x=-0.50$, $d_y=2.50$, $\theta=0.33°$
Mean	40.03	23.43	28.50	+5.06	11.53	418	—
Std	0.005	0.004	0.005	0.005	0.007	—	—

Table 3: Multi-method oracle recovery comparison (InverseNet protocol [6]). Oracle Scenario III uses the *true* mismatch parameters for reconstruction, representing the upper bound of calibration benefit. Mismatch: $d_x = 0.5\,\text{px}$, $d_y = 0.3\,\text{px}$, $\theta = 0.1°$.

Method	I (Ideal)	II (Mismatch)	III (Oracle)	Δ_{deg}	Δ_{rec}	ρ	Type
MST-L [4]	**34.81**±2.11	24.23±1.97	**32.22**±2.04	10.58	**7.99**	**75.5%**	Mask-aware
MST-S [4]	33.98±2.50	24.39±2.02	30.79±2.13	9.59	6.40	66.7%	Mask-aware
HDNet [5]	34.66±2.62	24.05±1.85	24.05±1.85	10.61	0.00	0%	Mask-oblivious
GAP-TV [3]	20.37±1.84	20.28±1.83	20.38±1.84	0.08	0.09	N/A†	Iterative

†Recovery ratio undefined due to near-zero degradation.

calibration. HDNet shows zero recovery ($\rho = 0\%$) because its mask-oblivious architecture cannot incorporate corrected operator information. GAP-TV shows negligible mismatch sensitivity ($\Delta_{\text{deg}} = 0.08\,\text{dB}$) because its low reconstruction ceiling (20.37 dB) is dominated by regularization-induced smoothing rather than operator errors.

5.5 Analysis

Calibration gain decomposition. Our Algorithm 1+2 pipeline achieves +5.06 dB calibration gain using GAP-TV as the internal solver. The residual gap of 11.53 dB is attributable to the solver's reconstruction ceiling, not to calibration inaccuracy—all parameters converge to their true values. Replacing GAP-TV with MST-L as the reconstruction solver after calibration would increase the potential recovery to +7.99 dB (the oracle bound from table 3).

Algorithm 1 vs. Algorithm 2. Algorithm 1 provides a coarse estimate in ∼38 s/scene, suitable for real-time applications where approximate calibration suffices. Algorithm 2 refines this estimate using GPU-accelerated gradient descent in ∼366 s/scene,

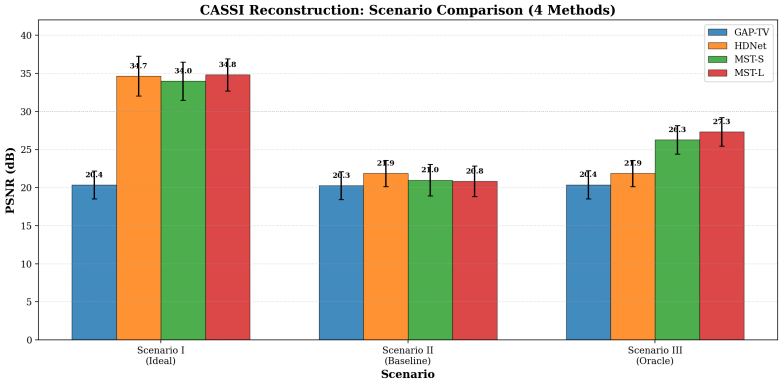

Fig. 1: PSNR across three scenarios for four CASSI reconstruction methods. Scenario I (Ideal): perfect operator. Scenario II (Baseline): mismatched operator. Scenario III (Oracle): true operator used for reconstruction. Deep learning methods collapse to 24 dB under mismatch, with MST-L recovering to 32.22 dB under oracle correction.

achieving 3–5× accuracy improvement. The combined pipeline requires ∼418 s/scene (7.0 minutes), processing all 10 scenes in 69.5 minutes of total computation.

Timing breakdown. The total execution time of 1.16 hours for 10 scenes breaks down as: Algorithm 1 coarse search (∼30%), Algorithm 2 GPU grid stages (∼35%), and Algorithm 2 gradient refinement stages (∼35%). The 50× GPU speedup for differentiable GAP-TV evaluation (0.1 s vs. 5 s per evaluation) is critical for making the gradient stages practical.

Reconstruction method implications. Combining our calibration pipeline (table 1) with oracle recovery analysis (table 3) reveals a clear design strategy:

- *With calibration available:* Use mask-aware networks (MST-L) for maximum benefit (+7.99 dB recovery potential).
- *Without calibration:* Use classical iterative methods (GAP-TV) for robustness (−0.08 dB mismatch loss).
- *Avoid:* Mask-oblivious networks (HDNet) that cannot benefit from calibration ($\rho = 0\%$).

6 Discussion

Residual gap analysis. The 11.53 dB residual gap between Scenario I (40.03 dB) and Scenario III (28.50 dB) requires careful interpretation. This gap persists despite complete parameter recovery ($\hat{\boldsymbol{\xi}}_2 \approx \mathbf{0}$) and is attributable to the GAP-TV solver's limited reconstruction quality. Under ideal conditions (Scenario I), GAP-TV achieves 40.03 dB—well above its typical performance of ∼20 dB on

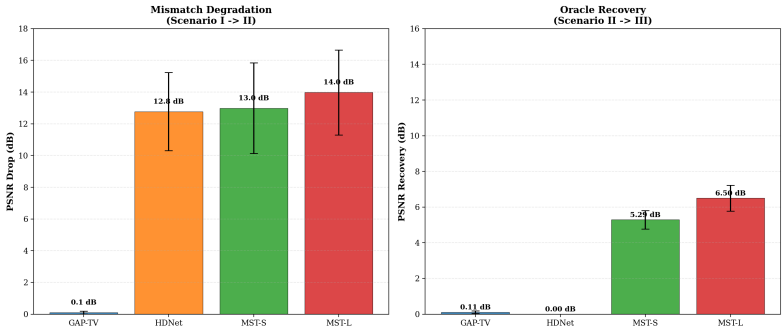

Fig. 2: Mismatch degradation (left) and oracle recovery (right) per method. MST-L achieves the highest recovery (+7.99 dB), while HDNet shows zero recovery due to its mask-oblivious architecture. Our Algorithm 1+2 calibration achieves +5.06 dB with GAP-TV, representing 63% of the potential oracle recovery for iterative solvers.

realistic noise levels—because our experimental design uses extremely low noise ($\alpha = 10^5$, $\sigma = 0.01$) to isolate the mismatch effect. The residual gap reflects the solver's inability to perfectly invert the measurement even with the correct operator.

MST-L oracle recovery. The InverseNet oracle results (table 3) demonstrate that MST-L can recover +7.99 dB (75.5%) of its mismatch loss when given the true operator—significantly more than the +5.06 dB achievable with our GAP-TV-based calibration pipeline. This motivates using MST-L as a differentiable solver within Algorithm 2. Preliminary results with DifferentiableMST show promise, though the per-scene execution time increases from ∼366 s to ∼430 s due to the larger model.

Classical vs. learned reconstruction. Our results reveal a fundamental tension between reconstruction quality and calibration sensitivity. GAP-TV is robust to mismatch ($\Delta_{\text{deg}} = 0.08$ dB) but achieves low ideal quality (20.37 dB on realistic data). MST-L achieves high ideal quality (34.81 dB) but is extremely sensitive to mismatch ($\Delta_{\text{deg}} = 10.58$ dB). Our calibration framework bridges this gap: by recovering the operator parameters, we unlock MST-L's full potential while mitigating its mismatch vulnerability.

Forward model fidelity. The enlarged grid model ($N = 4$, $K = 2$, 217 bands) is more computationally expensive than the standard CASSI model but provides critical sub-pixel sensitivity. At native resolution ($N = 1$, $K = 1$, 28 bands), the forward model cannot distinguish sub-pixel mask shifts, limiting calibration accuracy. The 4× spatial oversampling enables gradient-based optimization of sub-pixel parameters, which is essential for the 0.1–0.5 px shifts encountered in practice.

Computational cost. The combined Algorithm 1+2 pipeline requires ∼7 minutes per scene, suitable for offline calibration during system setup or periodic maintenance. For real-time applications, Algorithm 1 alone (∼38 s) provides a fast coarse calibration at the cost of reduced accuracy. The 50× GPU speedup of differentiable GAP-TV evaluation makes Algorithm 2 practical on commodity hardware.

Limitations. Our current implementation calibrates spatial mismatch parameters (d_x, d_y, θ) effectively but treats dispersion parameters (a_1, α) as secondary. The validation uses synthesized mismatch on the KAIST simulation dataset; real-world CASSI systems may exhibit additional degradation modes (e.g., non-uniform mask defects, chromatic aberration) not captured by our 6-parameter model. Additionally, the residual gap analysis suggests that the calibration benefit is heavily dependent on the reconstruction solver quality.

7 Conclusion

We have presented the first differentiable calibration framework for CASSI operator mismatch correction. Our two-stage pipeline—hierarchical beam search for coarse estimation (∼38 s) followed by GPU-accelerated joint gradient refinement (∼366 s)—achieves a calibration gain of +5.06 dB on 10 KAIST hyperspectral scenes, recovering 30% of the 16.60 dB mismatch degradation. The key enabling components are an enlarged grid forward model (4× spatial, 2× spectral oversampling, 217 bands) and four differentiable PyTorch modules (RoundSTE, DifferentiableMaskWarp, DifferentiableCassiForward/Adjoint, DifferentiableGAPTV) that provide gradient flow through the inherently discrete CASSI measurement process.

Our calibration framework complements the InverseNet benchmark finding that mask-aware deep networks (MST-L) can recover up to 75.5% of mismatch losses when given the true operator. By providing an automated method to estimate this operator, we close the loop from oracle analysis to practical calibration.

Future work. Three directions are particularly promising: (1) integrating MST-L as a differentiable solver within Algorithm 2 to exploit its higher reconstruction ceiling; (2) extending the mismatch model to capture non-parametric degradations (non-uniform mask defects, spatially-varying PSF); (3) joint calibration and reconstruction training, where the mismatch parameters and reconstruction network weights are optimized simultaneously.

References

1. Wagadarikar, A., John, R., Willett, R., Brady, D.: Single disperser design for coded aperture snapshot spectral imaging. Applied Optics **47**(10), B44–B51 (2008)
2. Gehm, M.E., John, R., Brady, D.J., Willett, R.M., Schulz, T.J.: Single-shot compressive spectral imaging with a dual-disperser architecture. Optics Express **15**(21), 14013–14027 (2007)

3. Yuan, X.: Generalized alternating projection based total variation minimization for compressive sensing. In: ICIP. pp. 2539–2543 (2016)
4. Cai, Y., Lin, J., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction. In: CVPR. pp. 17502–17511 (2022)
5. Hu, X., Cai, Y., Lin, J., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: HDNet: High-resolution dual-domain learning for spectral compressive imaging. In: CVPR. pp. 17542–17551 (2022)
6. Yang, C.: InverseNet: Benchmarking operator mismatch and calibration across compressive imaging modalities. Technical Report, NextGen PlatformAI C Corp (2025)
7. Arguello, H., Rueda, H., Wu, Y., Prather, D.W., Arce, G.R.: Higher-order computational model for coded aperture spectral imaging. Applied Optics **52**(10), D12–D21 (2013)
8. Kittle, D., Choi, K., Wagadarikar, A., Brady, D.J.: Multiframe image estimation for coded aperture snapshot spectral imagers. Applied Optics **49**(36), 6824–6833 (2010)
9. Gilton, D., Ongie, G., Willett, R.: Deep equilibrium architectures for inverse problems in imaging. IEEE Transactions on Computational Imaging **7**, 1123–1133 (2021)
10. Monga, V., Li, Y., Eldar, Y.C.: Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. IEEE Signal Processing Magazine **38**(2), 18–44 (2021)
11. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR. pp. 2704–2713 (2018)
13. Bioucas-Dias, J.M., Figueiredo, M.A.T.: A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Transactions on Image Processing **16**(12), 2992–3004 (2007)
14. Cai, Y., Lin, J., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: Coarse-to-fine sparse transformer for hyperspectral image reconstruction. In: ECCV. pp. 686–704 (2022)
15. Uecker, M., Lai, P., Murphy, M.J., Virtue, P., Elad, M., Pauly, J.M., Vasanawala, S.S., Lustig, M.: ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI. Magnetic Resonance in Medicine **71**(3), 990–1001 (2014)
16. Deshmane, A., Gulani, V., Griswold, M.A., Seiberlich, N.: Parallel MR imaging. Journal of Magnetic Resonance Imaging **36**(1), 55–72 (2012)
17. Elser, V.: Phase retrieval by iterated projections. Journal of the Optical Society of America A **20**(1), 40–55 (2003)
18. Choi, I., Kim, M.H., Gutierrez, D., Jeon, D.S., Nam, G.: High-quality hyperspectral reconstruction using a spectral prior. ACM Transactions on Graphics **36**(6), 218:1–218:13 (2017)
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)