SUPPLEMENTARY MATERIAL: FLOW-MATCHING GUIDED DEEP UNFOLDING FOR HYPERSPECTRAL IMAGE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

1 DERIVATION OF CONDITIONAL OT OBJECTIVE

In the main paper, we directly introduced the regression objective for flow matching, parameterizing the conditional vector field $v_\theta(x_t, t, y)$ and learning it by least-squares regression. For completeness, here we provide a detailed derivation showing how the simplified objective arises from the conditional optimal transport (OT) formulation under a constant-velocity ODE. Specifically, we start from the dynamic OT problem (Benamou & Brenier, 2000), connect it to displacement interpolation, and show how this naturally leads to the constant velocity target $x_1 - x_0$. This derivation clarifies how the ODE trajectory definition yields the training loss presented in Eqs. (10)–(11) of the main text.

1.1 CONDITIONAL OPTIMAL TRANSPORT VECTOR FIELD DERIVATION

Let y denote the conditioning signal (e.g., a CASSI measurement). We consider two conditional distributions on $\mathbb{R}^d$: the source μ_0^y with density $p_0(x \mid y)$ and the target μ_1^y with density $p_1(x \mid y)$. A time-dependent conditional density μ_t^y transported by a velocity field $v_t^y(x)$ satisfies the (conditional) continuity equation

$$\partial_t \mu_t^y(x) + \nabla \cdot \big(\mu_t^y(x) v_t^y(x)\big) = 0, \qquad t \in [0,1], \tag{1}$$

with boundary conditions μ_0^y, μ_1^y.

In the *dynamic OT* formulation (Benamou & Brenier, 2000), the 2-Wasserstein geodesic between μ_0^y and μ_1^y is obtained by minimizing the conditional kinetic energy

$$\inf_{\{\mu_t^y, v_t^y\}} \int_0^1 \int \tfrac{1}{2}\|v_t^y(x)\|_2^2 \, \mu_t^y(x)\, dx\, dt \quad \text{s.t.} \quad \begin{cases} \partial_t \mu_t^y + \nabla \cdot (\mu_t^y v_t^y) = 0, \\ \mu_0^y, \mu_1^y \text{ given.} \end{cases} \tag{2}$$

When a (conditional) OT map (Villani et al., 2008) $T_y : \mathbb{R}^d \to \mathbb{R}^d$ exists, the geodesic curve is the *displacement interpolation* (McCann, 1997)

$$X_t = (1-t)X_0 + t\, T_y(X_0), \qquad X_0 \sim \mu_0^y, \;\; t \sim \mathrm{Unif}[0,1], \tag{3}$$

with interpolated law $\mu_t^y = \big((1-t)\mathrm{Id} + t\, T_y\big)_{\#}\mu_0^y$. Along each sample path $\gamma(t) = (1-t)x_0 + t x_1$, the geodesic velocity is constant:

$$\dot\gamma(t) = x_1 - x_0, \qquad \forall\, t \in [0,1]. \tag{4}$$

Thus the (conditional) ground-truth vector field is

$$v^\star(x_t, t, y) = x_1 - x_0, \quad \text{whenever } x_t = (1-t)x_0 + t x_1, \;\; (x_0, x_1) \sim \pi_y, \tag{5}$$

where π_y is an (optimal) coupling between μ_0^y and μ_1^y.

1.2 FROM FLOW MATCHING REGRESSION TO THE CONSTANT-VELOCITY TARGET

Flow matching (FM) learns a parametric conditional field $v_\theta(x_t, t, y)$ by least-squares regression:

$$\hat\theta = \arg\min_\theta \; \mathbb{E}_{t, x_t \sim \mu_t^y} \big[\|v^\star(x_t, t, y) - v_\theta(x_t, t, y)\|_2^2\big]. \tag{6}$$

The sampling procedure consists of three main steps. First, a time variable t is uniformly sampled from the interval $[0, 1]$. Next, a pair (x_0, x_1) is drawn from the coupling distribution π_y. Finally, the intermediate state x_t is obtained by linearly interpolating between the two samples, that is, $x_t = (1-t)x_0 + tx_1$. Under the constant-velocity ODE $\dot{x}_t = x_1 - x_0$, the target is $x_1 - x_0$. Hence the FM loss reduces to

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{t,(x_0,x_1)\sim\pi_y}\left[\|(x_1 - x_0) - v_\theta((1-t)x_0 + tx_1, t, y)\|_2^2\right], \tag{7}$$

which matches the simplified loss used in our main text.

1.3 Interpretation

At optimum, the regressor satisfies

$$v_\theta^\star(x_t, t, y) = \mathbb{E}[x_1 - x_0 \mid x_t, t, y], \tag{8}$$

i.e., the learned field is the conditional mean velocity of the OT geodesic. This connects flow matching training to estimating the OT vector field, ensuring theoretical consistency. Our proposed *mean-velocity loss* further enforces this property, improving global consistency when couplings are noisy.

2 Extra Experimental Settings of Our Full Model

Stage design. We follow an unfolding design with 10 stages. Parameter sharing across stages is enabled, consistent with previous LADE-DUN (Wu et al., 2024). To further enhance the learned prior, we integrate flow matching with a mean velocity regularization weight of $\lambda_{\text{reg}} = 5$.

Training protocol. We train 10-stage models with a batch size of 1 and patch size of 256×256. In both training phases, the maximum epoch is set to 300 and the learning rate is initialized at 4×10^{-4}. Gradient clipping is enabled to stabilize training. We employ the MultiStepLR scheduler from Paszke et al. (2019) with milestones at epochs $\{50, 100, 150, 200, 250\}$ and decay factor $\gamma = 0.5$.

Initialization and ODE solver. We adopt Gaussian initialization with noise scale 1.0 and apply an ODE sampler based on the `rk45` solver (Dormand & Prince, 1980), with tolerance set to 10^{-5}.

These additional settings highlight the key practical configurations in our pipeline and provide useful reference for reproducing and extending our work.

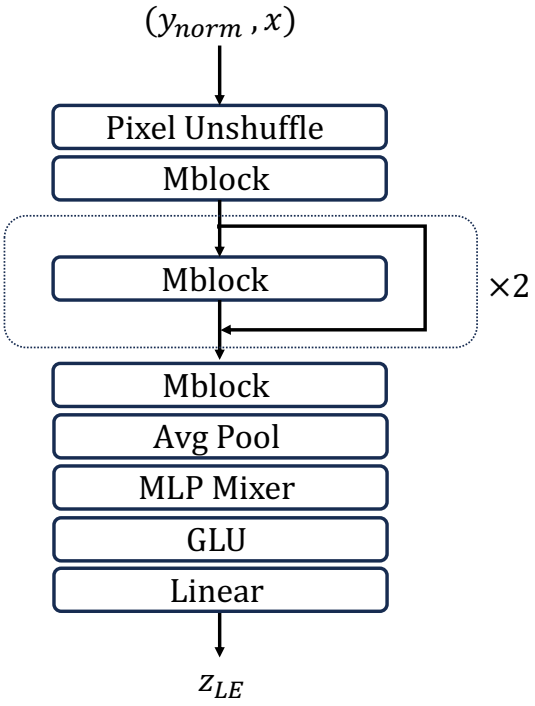
3 Theoretical Insight Explanation and Detailed Model Architecture

3.1 On theoretical insight of using flow matching prior

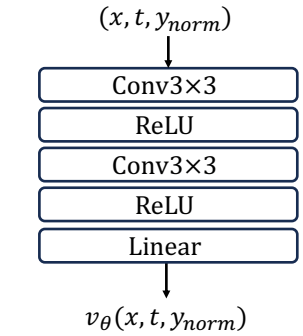
We acknowledge that the manuscript emphasizes empirical evidence. The key insight is that flow matching enables high-quality latent priors without the heavy sampling overhead inherent to diffusion. In LADE-DUN, the number of diffusion steps must be aggressively reduced (e.g., 16 steps) to maintain feasible inference cost, which inevitably compromises the generated prior. In contrast, our flow-matching module supports RK-based sampling without step-count restrictions and therefore can freely increase sampling precision while keeping the runtime comparable to diffusion-based few-step priors. This capability directly improves prior quality and consequently reconstruction accuracy. The mean-velocity constraint further enhances this advantage by encouraging a globally consistent vector field, leading to more stable training and improved hyperspectral consistency.

3.2 Latent Encoder

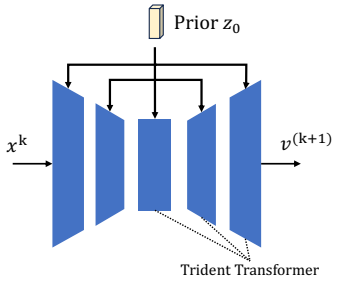
The latent encoder (**LE**) follows the structure illustrated in Fig. 1a. Given the normalized measurement y_{norm} and input x, the encoder first applies a **Pixel Unshuffle** operation to rearrange spatial information into the channel dimension, enabling more compact feature extraction. This is followed by a sequence of stacked **Mblocks** (Howard et al., 2019), where local and global spatial dependencies are progressively modeled. To increase representational capacity while maintaining efficiency,



(a) Structure of Latent Encoder (LE).



(b) Structure of Simple CNN in flow matching to predict velocity.



(c) U-shaped denoiser in FMU.

Figure 1: Detailed structures of LE, velocity predictor CNN in Flow Matching and denoiser in FMU.

two **Mblocks** are repeated in a loop as indicated in the figure. Subsequently, the features are further refined through an additional **Linear** layer, before being aggregated by an **average pooling**.

On top of the pooled features, the encoder employs an *MLP Mixer* and a *Gated Linear Unit (GLU)* to enhance channel interactions and improve non-linear feature fusion. Finally, a *linear projection* maps the encoded representation into the latent vector z, which serves as the compact and informative embedding for subsequent stages of our framework. This design ensures a balance between efficiency and expressive power, providing a strong latent representation tailored for HSI.

3.3 Denoiser in FMU

We use a U-shaped Denoiser which consists of Trident Transformer (Wu et al., 2024). The structure of denoiser is shown in Fig. 1c, which incorporates different scales of Trident Transformer.

3.4 Trident Transformer

we use Trident Transformer from Wu et al. (2024) to effectively aggregate high-quality degradation-free prior knowledge from flow matching for compensation. The overview of Trident Transformer assisted with flow-matching prior is in Fig. 2. The input of the transformer is U_i, in which i means that the Trident Transformer is of the i-th scales. The input is first divided to U_i^S and U_i^C by channels to extract different information. Then either is sent to Spectral Flow or Cross Spectral&Prior Flow. Flow-matching prior is sent to Cross Prior Flow to assist denoising. Inside each flow there are convolutions and QKV calculations. Refer to Wu et al. (2024) for more details.

3.5 Velocity Predictor in Flow Matching

As for the network used to predict velocity field in flow matching, we adopt Simple CNN after ablations. Detailed structure is shown in Fig. 1b.

4 Additional Results and Training pseudocode

4.1 Simulation Experiment Visualizations

We first show additional simulation experiment results to highlight the ability of our method to recover spatial details and spectral structures (see Fig. 3). These visualizations complement the main

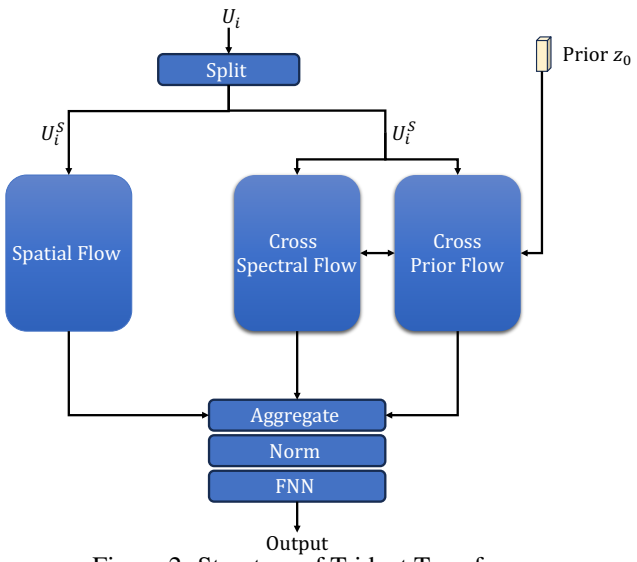


Figure 2: Structure of Trident Transformer.

text and provide a more comprehensive view of the reconstruction quality achieved by our approach. They confirm that our method produces sharper edges and more consistent spectral responses compared with competing with previous works.

4.2 Two-Phase Training Procedure

Besides visual results, we present the two-phase training procedure of our model. In Phase 1, the latent encoder and unfolding module are jointly trained for accurate recovery from measurements. In Phase 2, flow matching is introduced to learn a generative prior, further enhancing reconstruction quality and robustness. The pseudocode for these two stages is summarized in Algorithm 1, 2. During inference, we set $I_E = (y_{norm}, 0)$ instead of (y_{norm}, x_{gt}) o ensure that no ground-truth information is used.

4.3 Full quantitative CASSI results

In addition to the results reported in the main paper, we also provide full quantitative evaluations on the CASSI benchmark and additional visual comparisons across more scenes. (See table 3)

4.4 Evaluation metrics, fairness, and efficiency of comparison

To further assess generalization beyond PSNR and SSIM, we additionally report results on three widely used hyperspectral quality metrics: SAM and ERGAS, which are specifically designed to evaluate spectral fidelity of HSI reconstructions, and MANIQA, which measures perceptual quality in the RGB space. In particular, SAM quantifies the average spectral angle between reconstructed and ground-truth spectra, and ERGAS evaluates the global relative reconstruction error across all bands. MANIQA is computed by projecting the hyperspectral cube into the RGB space and applying a no-reference IQA model to the resulting pseudo-RGB image. As shown in Table 1, FMU achieves the best performance on four out of five metrics, significantly improving PSNR, SSIM, SAM, and ERGAS over all baselines. Although LADE-DUN is marginally better in MANIQA (0.6294 vs. 0.6293), this difference is negligible, whereas our gains on hyperspectral-specific metrics (SAM, ERGAS) are substantial. These results demonstrate that our evaluation goes beyond PSNR/SSIM and that FMU consistently offers better spectral fidelity and overall reconstruction quality under a fair and comprehensive comparison protocol. Beyond reconstruction quality, we also compare the efficiency of FMU with the diffusion-prior baseline LADE-DUN under a matched unfolding depth. Table 2 reports PSNR, FLOPs, and wall-clock inference time. Under a very similar FLOPs budget (96.69G vs. 98.84G), FMU improves PSNR from 40.97 dB to 42.13 dB (+1.16 dB) while reducing inference time from 0.521 s to 0.390 s (about 1.3$\times$ faster). This indicates that, despite slightly higher

Method	PSNR (dB)↑	SSIM↑	SAM (rad)↓	ERGAS↓	MANIQA↑
BIRNAT	38.76	0.9792	0.1064	16.79	0.6119
BiSRNet	35.84	0.9524	0.2126	23.12	0.5714
CST-L-Plus	37.73	0.9762	0.1380	19.21	0.6180
DAUHST-9stg	38.81	0.9815	0.1120	17.05	0.6273
DGSMP	32.99	0.9471	0.1688	33.57	0.5664
MST++	39.21	0.9825	0.0989	16.25	0.6256
TSA-Net	37.68	0.9759	0.1103	18.82	0.6124
SPECAT	40.37	0.9860	0.0877	14.04	0.6292
LADE-DUN-10stg	40.97	0.9882	0.0708	12.70	**0.6294**
FMU (Ours)	**42.13**	**0.9900**	**0.0651**	**11.26**	0.6293

Table 1: Multi-metric comparison on the optical-filter-based system. FMU achieves the best performance on four out of five metrics.

Method	PSNR (dB)↑	FLOPs (G)	Inference Time (s)↓
LADE-DUN-10stg	40.97	**96.69**	0.521
FMU (Ours)	**42.13**	98.84	**0.390**

Table 2: Efficiency comparison with the diffusion-based LADE-DUN under a similar computational budget. FMU achieves both higher PSNR and lower inference latency.

theoretical FLOPs, the flow-matching prior yields a strictly better quality–efficiency trade-off than the diffusion prior in LADE-DUN.

4.5 Additional experiments on KAIST and ICVL (zero-shot generalization)

To further evaluate robustness beyond the training distribution, we additionally conduct experiments on two unseen real-world datasets: KAIST and ICVL. Unlike the main benchmark, which follows the standard optical-filter-based setting, these two datasets contain different spectral characteristics and scene distributions, making them suitable for testing zero-shot generalization. **KAIST.** FMU preserves its advantage on previously unseen KAIST scenes, improving PSNR (43.44 vs. 43.04) and reducing ERGAS (13.52 vs. 14.09), indicating more accurate hyperspectral reconstruction and stronger spectral consistency. **ICVL.** The performance gap becomes more pronounced on ICVL. FMU improves PSNR by +1.66 dB and also yields clearly lower SAM and ERGAS. This suggests that FMU is more robust when the degradation process and spectral statistics differ from those observed during training. Overall, these results show that FMU consistently maintains its superiority without any fine-tuning on new datasets. This supports that flow-matching priors generalize more reliably than diffusion-based priors within physics-guided unfolding frameworks.

References

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 2000.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 1980.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019.

Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 1997.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Method	PSNR (dB)	SSIM
TwIST	23.12	0.669
DNU	30.74	0.863
MST++	35.99	0.951
BIRNAT	37.58	0.960
LRSDN	36.08	0.938
DAUHST-9stg	38.36	0.967
DADF-Plus-3	39.28	0.972
PADUT-5stg	37.84	0.967
RDLUF-MixS2-3stg	37.56	0.963
LADE-DUN-3stg	38.31	0.972
PADUT-12stg	38.89	0.974
RDLUF-MixS2-9stg	39.57	0.974
LADE-DUN-10stg	40.09	0.979
FMU	**41.02**	**0.988**

Table 3: CASSI reconstruction results of different methods.

More scenes from KAIST and ICVL				
Method	PSNR (dB)↑	SSIM↑	SAM (rad)↓	ERGAS↓
More scenes from KAIST				
FMU (Ours)	**43.44**	**0.9942**	0.0645	**13.52**
LADE-DUN	43.04	0.9939	**0.0638**	14.09
Scenes from ICVL				
FMU (Ours)	**33.58**	**0.9492**	**0.1046**	**73.20**
LADE-DUN	31.92	0.9396	0.1264	81.80

Table 4: Zero-shot reconstruction results on additional KAIST and ICVL scenes.

Algorithm 1: Phase 1: Learn Prior and Reconstruction

Data: Ground-truth HSI $\boldsymbol{x}$, Measurement $\boldsymbol{y}$, Sensing matrix $\mathbf{A}$, Unfolding Network ϕ_{FMU}
Result: Trained Latent Encoder LE, Trained Unfolding Network ϕ_{FMU}

$\boldsymbol{y}_{norm} \leftarrow \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{y}$;
$I_E \leftarrow \text{concat}(\boldsymbol{y}_{norm}, \boldsymbol{x})$;
for *epoch* $\leftarrow 1$ **to** E **do**
  Shuffle dataset;
  for *each batch* B *in training data* **do**
    $I_E \leftarrow \text{concat}(\boldsymbol{y}_{norm}, \boldsymbol{x})$;
    $\boldsymbol{z}_{LE} \leftarrow LE(I_E)$;
    $\hat{\boldsymbol{x}} \leftarrow \phi_{FMU}(\boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{z}_{LE})$;
    $\mathcal{L}_1 \leftarrow ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_1$;
    Update parameters $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_1$;
  end
end

Cédric Villani et al. *Optimal transport: old and new*. Springer, 2008.

Zongliang Wu, Ruiying Lu, Ying Fu, and Xin Yuan. Latent diffusion prior enhanced deep unfolding for snapshot spectral compressive imaging. In *ECCV*, 2024.

Algorithm 2: Phase 2: Flow Matching for Prior Generation

Data: Measurement $\boldsymbol{y}$, Sensing matrix $\mathbf{A}$, Encoder LE, Unfolding Network ϕ_{FMU}, Flow Matching denoise network ϕ_{FM}

Result: Trained ϕ_{FMU}, Trained ϕ_{FM}

$\boldsymbol{y}_{norm} \leftarrow \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{y}$;

$I_E \leftarrow \text{concat}(\boldsymbol{y}_{norm}, \boldsymbol{x})$;

Freeze encoder LE from Phase 1;

for $epoch \leftarrow 1$ **to** E **do**

  Shuffle dataset;

  for *each batch* B *in training data* **do**

    $I_E \leftarrow \text{concat}(\boldsymbol{y}_{norm}, \boldsymbol{x})$;

    $\boldsymbol{z}_{LE} \leftarrow LE(I_E)$;

    $\hat{\boldsymbol{z}}_{FM} \leftarrow \phi_{FM}(I_E)$;

    $\hat{\boldsymbol{x}} \leftarrow \phi_{FMU}(\boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{z}_E)$;

    $\mathcal{L}_2 \leftarrow ||\hat{\boldsymbol{z}}_{FM} - \boldsymbol{z}_{LE}||_2^2 + \mathcal{L}_{rec}$;

    Update parameters $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_2$;

  end

end

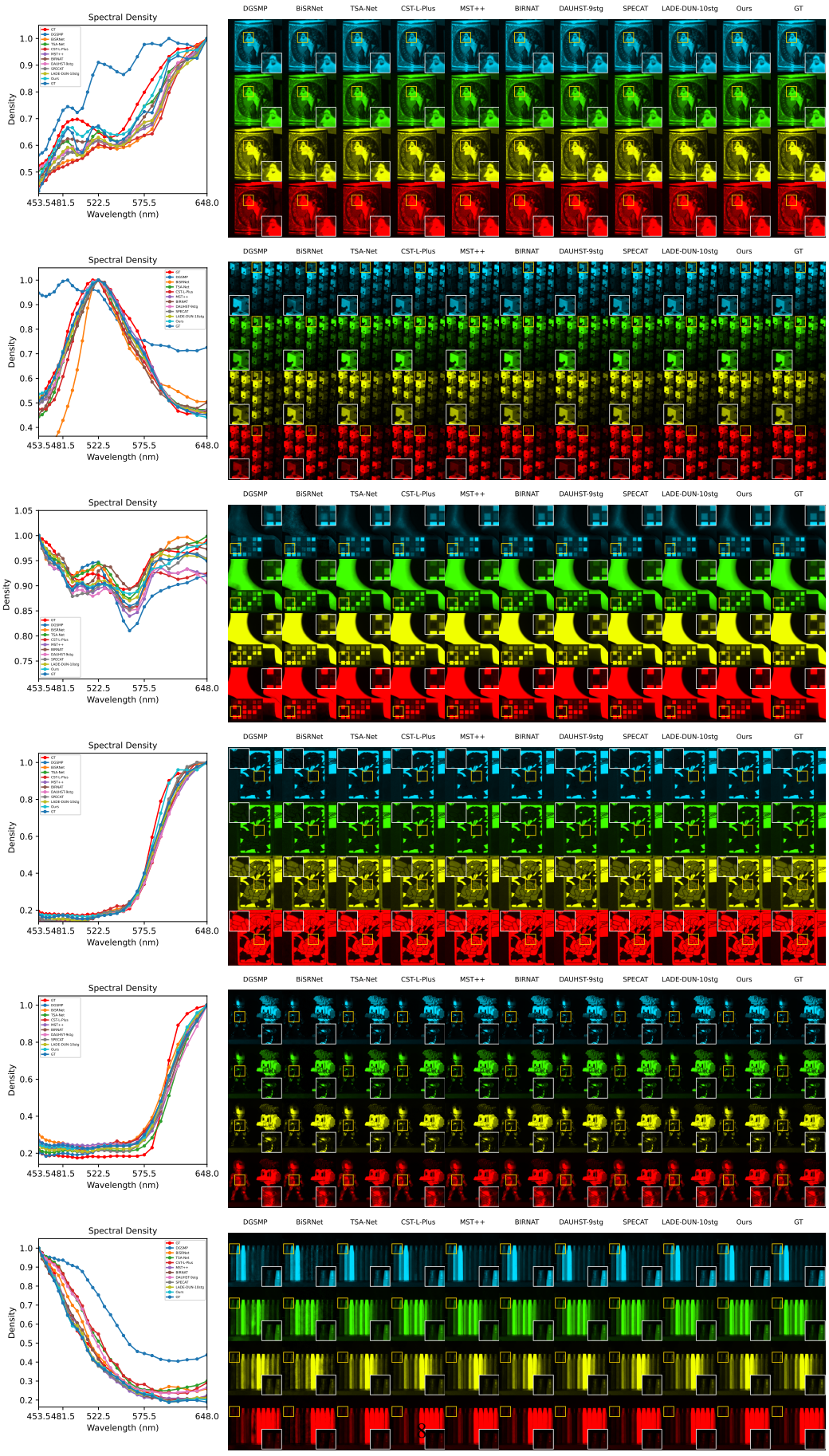


Figure 3: Qualitative comparisons across test cases. Each row shows results on a different example.