

LCPNet: Latent Consistent Proximal Unfolding Network for Infrared Small Target Detection

Tianfang Zhang, Fengyi Wu, Lei Li, Chang Liu, Zhenming Peng, Huaping Zhang, Xiangyang Ji

Abstract—Infrared small target detection (IRSTD) aims to identify long distance small targets from complex infrared backgrounds, and is a fundamental task in remote sensing. Deep learning methods have improved IRSTD by learning discriminative image-to-mask mappings, but such feed-forward designs often underuse physical decomposition structure between targets and backgrounds. Deep unfolding methods partially address this issue by embedding model-driven iterations into neural networks, yet existing designs still operate mainly in image domain and use updates and memory mechanisms that are not fully coupled with underlying optimization process. To address these limitations, we propose Latent Consistent Proximal unfolding network (LCPNet). First, we verify that low-rank prior remains valid in latent representations and perform unfolding in this space, preserving physical constraint while avoiding repeated compression of intermediate states. Second, we derive a Latent Consistent Proximal (LCP) solver that evolves each latent variable from its previous state rather than reconstructing through an indirect residual, and stabilizes small target updates through task-adaptive normalization and gain control. Third, we introduce Shared Optimization Memory (SOM), a common historical state shared by all decomposition variables to provide coordinated guidance across unfolding stages. Extensive experiments on four public benchmarks demonstrate that LCPNet outperforms state-of-the-art methods while achieving accurate and robust detection with low false alarms and competitive efficiency. Model and code are available at <https://github.com/Tianfang-Zhang/LCPNet>.

Index Terms—Infrared small target detection, deep unfolding network, interpretability, low-rank and sparse decomposition, image segmentation.

I. INTRODUCTION

INFRARED small target detection (IRSTD) is a fundamental capability for long-range perception systems, with broad value in remote sensing monitoring, maritime and aerial early warning, and security reconnaissance [1], [2]. Compared to other imaging techniques, infrared sensors capture thermal radiation and can operate under weak illumination, nighttime conditions, and high disturbance. This passive imaging mechanism renders IRSTD uniquely effective in scenarios where it is difficult to observe targets through conventional visual cues. Leveraging these advantages, IRSTD offers significant practi-

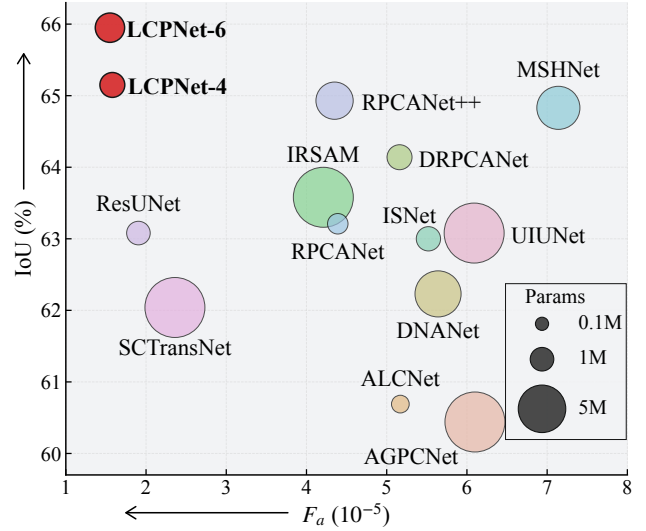


Fig. 1. Comparison of $F_a(10^{-5})$ -IoU scatter plots for infrared small target detection algorithms on IRSTD-1k [7]. Circle size indicates the parameter number. Points closer to top-left indicate better performance.

cal value and is one of the most prominent and challenging tasks in low-level vision and remote sensing.

The difficulty of IRSTD stems from the inherent imaging characteristics of infrared small targets. In long-range observations, a target often occupies only a few pixels, lacks stable shape and texture, and appears as an extremely sparse response embedded in a large background region [1], [2]. Such targets provide insufficient semantic evidence for conventional object detection pipelines. Meanwhile, infrared backgrounds are often highly nonuniform: cloud edges, sensor noise, and other locally bright clutter may exhibit target-like contrast, leading to false alarms [3]. Therefore, IRSTD relies more on reliable background suppression, weak target enhancement, and the precise distinction between sparse target responses and residual clutter signals [4]–[6].

Recent deep learning methods have substantially advanced IRSTD by learning more powerful multiscale and context-aware representations from annotated infrared images. CNN-based segmentation models, U-shaped architectures, attention mechanisms, and dense nested networks have improved target localization and background discrimination under complex scenes [3], [6], [8]–[12]. These methods demonstrate that data-driven feature learning can compensate for the weak appearance of infrared small targets. However, most traditional deep learning models remain confined to an end-to-end learning paradigm that directly maps infrared images to target masks. Under this paradigm, the learned behavior is mainly shaped by

Corresponding author: Xiangyang Ji.

Tianfang Zhang, Chang Liu and Xiangyang Ji are with the Department of Automation, Tsinghua University, Beijing, 100190, China. (E-mail: sparkcar-leon@gmail.com; {liuchang2022, xyji}@tsinghua.edu.cn)

Fengyi Wu and Zhenming Peng are with the School of Information and Communication Engineering and the Laboratory of Imaging Detection and Intelligent Perception, University of Electronic Science and Technology of China, Chengdu, 610054, China. (E-mail: wufengyi98@163.com, zm-peng@uestc.edu.cn)

Lei Li and Huaping Zhang are with the School of Artificial Intelligence, Beijing Institute of Technology, Beijing, 100081, China. (E-mail: {lilei, kevinzhang}@bit.edu.cn)

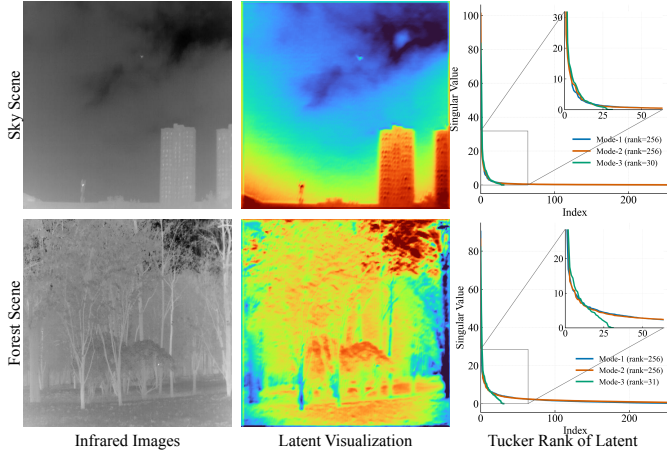


Fig. 2. Visualization of the low-rank property in latent domain. From top to bottom are the original infrared image, a visualized channel from the latent, and Tucker rank analysis of the latent. It can be observed that, whether in simple scene (sky *XDU82* in NUDT-SIRST [9]) or complex scene (forest *XDU41*), the latent exhibits distinct low-rank property in both visual visualization and Tucker rank analysis. For more scenes and rank analyses, please refer to Appendix D.

network architecture, loss supervision, and training data, rather than by an explicit physical model of infrared image formation. Consequently, it neglects the realistic physical modeling of background-target-noise decomposition. This limitation causes models to generate excessive and cumulative false positives in complex scenarios.

Physical modeling forIRSTD motivates a review of the model-driven methods. Infrared small target image naturally admits a low-rank and sparse decomposition formulation, where the observation can be regarded as the superposition of a structured background, sparse target, and noise component, i.e., $\mathbf{X} = \mathbf{B} + \mathbf{T} + \mathbf{N}$ as shown in Fig. 2. Classical RPCA and infrared patch-image models provide an interpretable foundation for separating sparse targets from low-rank backgrounds [5], [13]–[15]. However, traditional optimization-based methods rely on fixed handcrafted priors, require iterative matrix computations, and are sensitive to regularization parameters and background statistics. Deep unfolding offers a principled compromise by unrolling iterative optimization into a finite number of learnable stages, thereby combining interpretable algorithmic structure with data-driven representation learning [16]–[18]. This paradigm is particularly attractive forIRSTD because each stage can be associated with a target-background-noise separation process while still adapting the solver to real infrared data.

Despite this progress, existing deep unfolding methods still face three key limitations. First, to preserve the image-domain physical constraint, most unfolding frameworks transmit image-domain variables between adjacent stages [17], [18]. This design requires each stage to project its intermediate representation back to low-dimensional images. Such repeated compression preserves explicit physical interface, but it would weaken the high-frequency and low-contrast responses that are critical for small target detection. Second, existing learned proximal solvers often obtain variables through indirect resid-

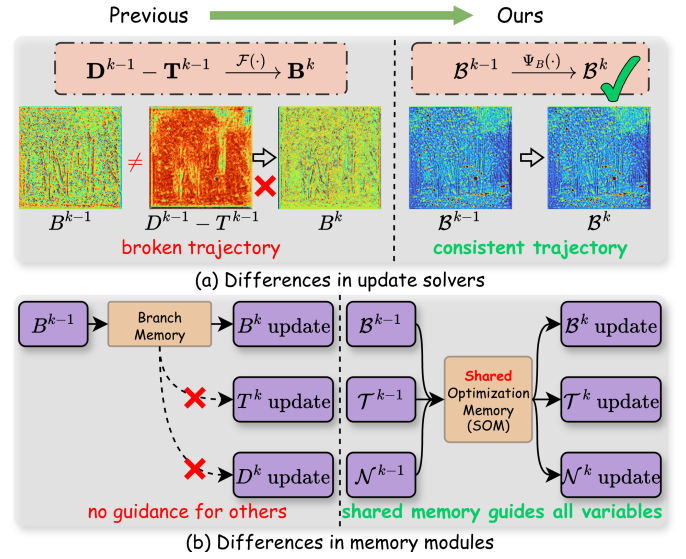


Fig. 3. Illustration of design differences between methods. (a) Upper: formula-based modeling. Lower: feature visualization. In previous studies (left), B^{k-1} and B^k are similar, but the update using $D^{k-1} - T^{k-1}$ violates feature consistency. In our LCP solver (right), the current state is directly determined by previous state, thereby ensuring feature consistency. (b). Previous methods (left) only used branch-level memory when updating B . Our SOM (right) features a shared optimization memory state for all variables, which directly participates in all updates, ensuring the historical memory of each one.

ual reconstruction, rather than directly evolving the previous state [16]–[18] as shown in Fig. 3(a). This weakens the continuity of background trajectory and may cause target leakage. Moreover, batch normalization is not well suited toIRSTD, since its statistics are dominated by abundant background pixels and fluctuate under small training batches, which can dilute weak target responses and destabilize repeated unfolding updates [19]–[22]. Third, existing cross-stage memory mechanisms are typically attached to a single decomposition variable, so historical information is reused only within a local update [18], [23]. Such branch-specific memory does not represent a shared optimization state of the unfolding process as in Fig. 3(b)(left), and therefore cannot provide common historical guidance for all coupled decomposition variables.

To address these issues, we propose Latent Consistent Proximal Unfolding Network (LCPNet) for infrared small target detection. After verifying that the low-rank physical prior remains valid in latent representation, we lift the execution domain of unfolding from image to latent, preserving the decomposition constraint while avoiding repeated projection of intermediate states. Then, LCPNet derives a Latent Consistent Proximal (LCP) solver for direct proximal updates, where each variable is evolved from its previous state rather than reconstructed from an indirect residual. Its updater further uses group-wise normalization and spectral gain control to better fit the small-batch and background-dominated characteristics ofIRSTD. Finally, LCPNet replaces branch-specific memory with Shared Optimization Memory (SOM), which is accessible to all decomposition variables and provides unified guidance for the coupled latent optimization across stages.

The main contributions of this work are summarized as follows:

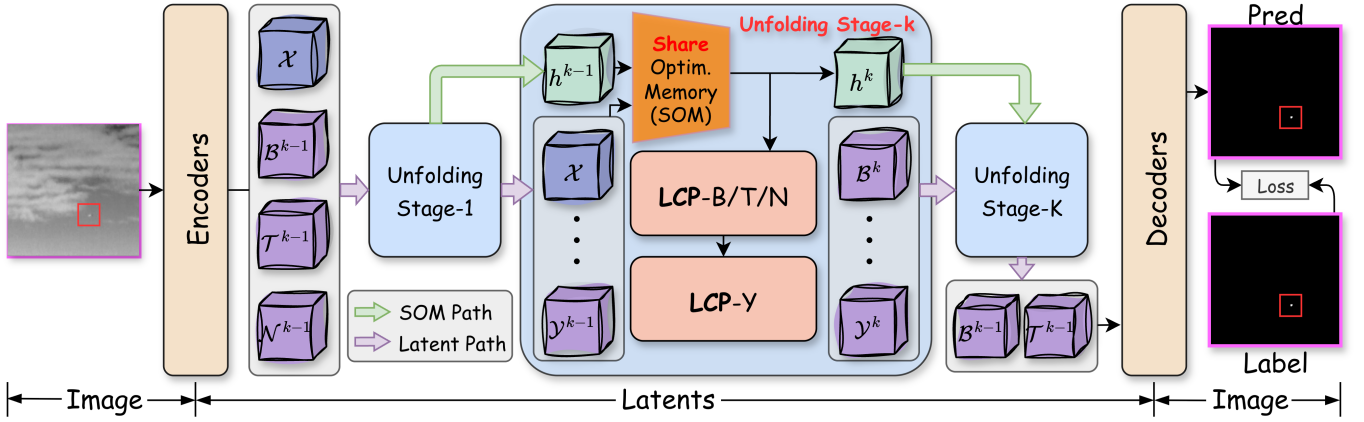


Fig. 4. Overall architecture of LCPNet. An infrared image is first lifted into the latent domain, where latent-space decomposition is performed through K unfolding stages, and the final target representation is decoded to image domain for detection.

- We verify that the low-rank decomposition prior remains valid in latent representations. Based on this observation, we formulate latent-domain unfolding to preserve the physical constraint while avoiding repeated variable compression.
- We derive a LCP solver for consistent proximal-state updates, allowing each latent variable to evolve from its previous state instead of indirect residual reconstruction. A group-wise and gain-controlled updater is further adopted to suppress background induced false alarms.
- We propose SOM as a system-level historical state for the unfolding process. SOM is shared by all decomposition variables and provides unified guidance for latent updates across stages.
- We propose LCPNet for infrared small target detection and extensive experiments on multipleIRSTD benchmarks demonstrate its superior detection accuracy, robust scene adaptability, and effective false-alarm suppression under diverse infrared scenes as shown in Fig. 1.

II. RELATED WORK

A. Infrared Small Targets Detection

ExistingIRSTD methods can be broadly grouped into Human Visual System (HVS)-based methods, optimization-based methods, and deep learning-based methods [1], [2], [24]. HVS-based methods are among the earliest solutions. They assume that small targets exhibit local intensity discontinuity relative to their surrounding background and therefore enhance target-like responses through handcrafted filters or local contrast measures. Representative methods include Top-hat filtering [25], max-mean and max-median filtering [26], local contrast measure (LCM) [4], and multiscale patch-based contrast measure (MPCM) [27]. These methods are simple, and computationally attractive, but their handcrafted saliency assumptions are often fragile in complex scenes.

Optimization-based methods regard infrared backgrounds as containing strong structural redundancy and can be represented by low-rank priors, while small targets are sparse outliers embedded in the observation. Based on this idea, the infrared patch-image (IPI) model formulates small target detection as a

low-rank and sparse decomposition problem [5]. NRAM further introduces non-convex rank approximation with structured sparsity to better characterize target components [14], while PSTNN exploits patch-tensor modeling with nonlocal priors for single-frame detection [15]. Compared with HVS-based methods, optimization-based methods provide clearer physical interpretability. However, they usually require iterative solvers, manually designed priors, and carefully tuned parameters, which limits their efficiency.

Deep learning-based methods have recently become the dominant paradigm forIRSTD. Most of these methods are non-unfolding networks, which directly learn a segmentation mapping from the infrared input to the target mask in an end-to-end manner. Early data-driven studies such as MDvsFA treatIRSTD as small object segmentation and consider the tradeoff between missed detections and false alarms [3]. Subsequent works improve representation learning by designingIRSTD-specific networks, including asymmetric contextual modulation [8], attentional local contrast learning [6], dense nested attention [9], UNet-in-UNet structures [10], attention-guided pyramid context aggregation [11], and shape-aware supervision [7]. More recent methods further explore scale and location sensitivity [12], SAM adaptation for infrared small targets [28], learnable saliency kernels [29], pinwheel-shaped convolution and scale-aware loss design [30], and low-level feature modeling [31]. These networks have greatly improved detection accuracy, but their inference process is usually a black-box input-to-mask mapping.

Another emerging branch is deep unfolding-basedIRSTD, which converts iterations of a traditional optimization algorithm into trainable stages [16]. This paradigm preserves model interpretability while replacing expensive analytical subsolvers with learnable neural modules. RPCANet and RPCANet++ instantiate this idea by unfoldingRPCA-like decomposition for infrared small target detection and enhancing it with learnable stage-wise modules [17], [18]. With its ability to integrate physical priors and data-adaptive learning, deep unfolding provides a promising foundation for more reliable and interpretableIRSTD models.

B. Deep Unfolding in Vision Models

Deep unfolding transforms an iterative optimization algorithm into a trainable network by mapping each iteration to a network stage and replacing fixed algorithmic components with learnable parameters or neural operators [16]. It bridges model-driven optimization and modern deep vision by preserving algorithmic structure and reducing iterative inference cost.

The idea first became influential in sparse coding and compressive sensing [32], [33]. ISTA-Net unfolds ISTA into a structured network with learnable transforms, achieving interpretable and efficient reconstruction [34], while LDAMP and AMP-Net unfold denoising-based approximate message passing for compressive image recovery [35], [36]. ADMM-Net similarly unfolds ADMM for MRI, replacing shrinkage functions with trainable counterparts [37]. Beyond this, unfolding has been widely adopted in image inverse problems [38], [39]. These methods avoid learning the entire inverse mapping from scratch; instead, they constrain certain calculations using known physical laws, while neural modules learn the missing prior [40], [41]. Low-rank and sparse decomposition is another natural domain for unfolding [13]. Learned RPCA extends this idea by unfolding scalable RPCA iterations and learning thresholds [42]. For IRSTD, RPCANet and RPCANet++ instantiate this principle and retain the background-target decomposition structure [17], [18].

Diffusion and score-based generative models can also be viewed as broad deep unfolding vision models, since they parameterize long denoising trajectories with learned stage-wise operators [43]–[49]. This broader view links unfolding, iterative generation, and inverse vision models through a shared emphasis on stage design, update stability, and state propagation.

C. Recurrent and State-Space Optimization in Vision

In unfolding vision networks, historical states help each stage reuse previous residuals along the optimization trajectory. This view is consistent with classical momentum and optimization [50], and is often implemented with recurrent modules such as LSTM, GRU, and ConvLSTM, which preserve temporal states through gated transitions [23], [51], [52].

Recurrent state modeling has been adopted in several inverse-vision solvers. Recurrent inference machines learn iterative inverse-problem updates with a hidden state [53] and MRI reuse cross-stage estimates through dense recurrent connections and recurrent variational networks [54]–[56]. These methods show that treating stage history as part of the solver state can improve reconstruction quality and optimization efficiency. State-space models further broaden this idea for visual representation. S4 and Mamba propagate hidden states over long sequences with efficient structured [57], [58], while vision mamba adapt such state propagation to images through bidirectional visual scans [59], [60]. Motivated by these developments, LCPNet uses a shared optimization memory to represent the historical trajectory of the coupled latent decomposition variables, allowing system-level state to guide all primal-variable updates across unfolding stages.

III. METHODOLOGY

This section presents the methodology of LCPNet in a progressive manner. We first verified that physical priors remain valid in latents and reformulate the decomposition in latent domain. We then derive LCP solver for direct proximal-state updates, followed by the SOM that provides historical guidance for all coupled variables. Finally, we describe how these components are instantiated in the network architecture.

A. Latent-Space Decomposition Formulation

Classical low-rank and sparse decomposition methods formulate IRSTD as an image-domain separation problem. Given an infrared observation $\mathbf{X}$, the background, target, and noise components are estimated directly in the physical image space:

$$\mathbf{X} = \mathbf{B} + \mathbf{T} + \mathbf{N}, \quad \text{s.t.} \quad \mathbf{X}, \mathbf{B}, \mathbf{T}, \mathbf{N} \in \mathbb{R}^{H \times W}. \quad (1)$$

This formulation is the basis of representative model-driven methods such as IPI, NRAM, and PSTNN [5], [14], [15], and it is also inherited by recent image-domain unfolding frameworks such as RPCANet and RPCANet++ [17], [18]. Its main advantage is that the optimization variables remain directly interpretable in the physical image domain. However, when this formulation is unfolded into a deep network, each stage still exchanges single-channel physical variables and repeatedly projects the optimization trajectory back to $\mathbb{R}^{H \times W}$. This low-dimensional stage interface can attenuate high-frequency small-target clues during reconstruction and prevents background, target, and noise representations from evolving persistently in a high-dimensional feature space.

To address these limitations, we first verified the low-rank property of latent variable through visualization and singular value analysis, as shown in Fig. 2. The singular values plot of the feature tensor decay rapidly, and indicates that latent representation still satisfies the low-rank physical assumption. This prior is applicable to both simple and complex scenarios, and Appendix D further demonstrates this prior from multiple perspectives. More specifically, we use a set of encoders to map image-domain variables to latent variables:

$$\begin{aligned} \mathcal{X} &= E_X(\mathbf{X}), \quad \mathcal{B} = E_B(\mathbf{B}), \\ \mathcal{T} &= E_T(\mathbf{T}), \quad \mathcal{N} = E_N(\mathbf{N}). \end{aligned} \quad (2)$$

The decomposition constraint is then imposed in the latent space:

$$\mathcal{X} = \mathcal{B} + \mathcal{T} + \mathcal{N}, \quad \text{s.t.} \quad \mathcal{X}, \mathcal{B}, \mathcal{T}, \mathcal{N} \in \mathbb{R}^{H \times W \times C}. \quad (3)$$

Compared with the image-domain formulation, the latent manifold provides a high-dimensional stage interface, allowing small-target and background structures to be propagated through C -channel optimization variables. This design preserves weak high-frequency target evidence across stages and lets background, target, and noise representations evolve on a richer latent manifold. After the iterative decomposition, projectors map the target-related latent variables back to the image domain for final detection output.

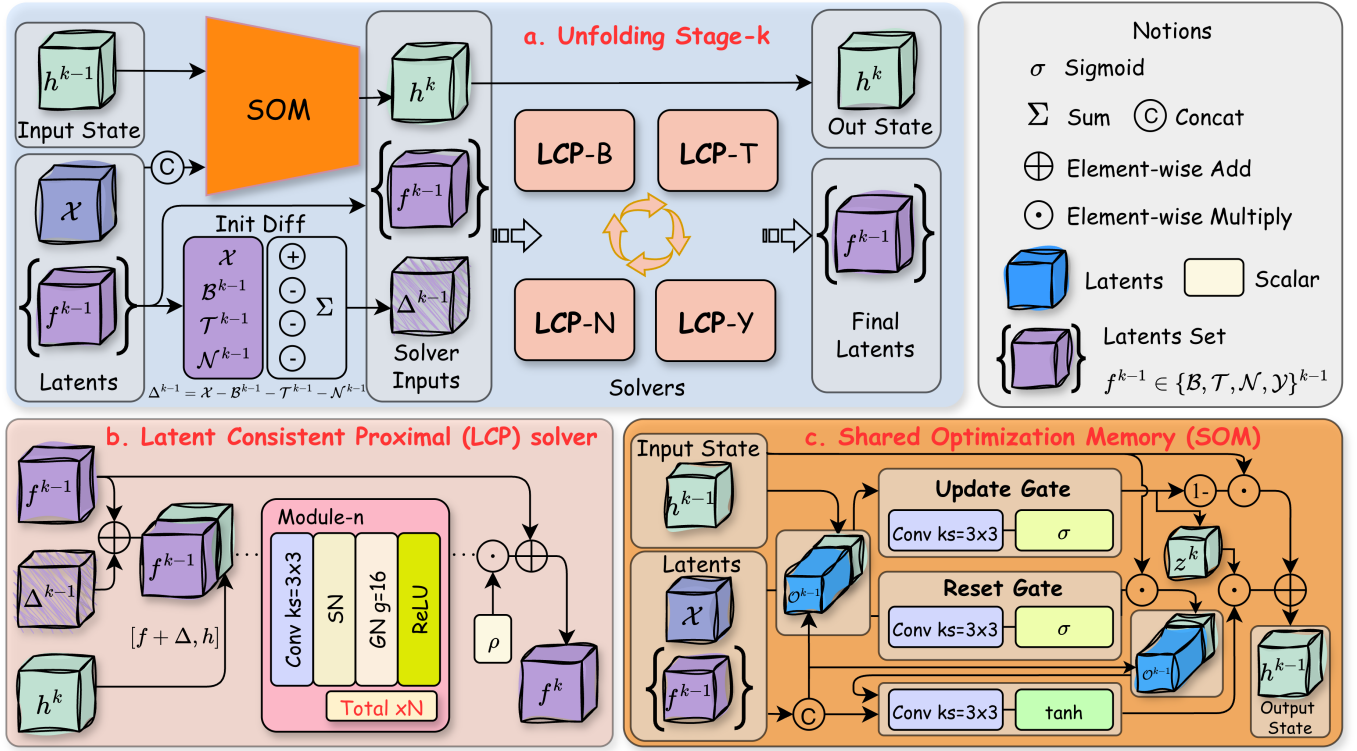


Fig. 5. Details of our custom-designed modules. (a) Illustration of the k -th unfolding stage, $f^{k-1} \in \{\mathcal{B}, \mathcal{T}, \mathcal{N}, \mathcal{Y}\}^{k-1}$ is a set of latent variables. (b) Illustration of LCP solver, each solver contains N convolutional modules. ks denotes convolutional kernel size, and g denotes the group number in GN. (c) Illustration of SOM, where $O^{k-1} = \text{Cat}([\mathcal{X}, \{f^{k-1}\}])$.

Under this latent-space physical model, the general decomposition objective of IRSTD can be written as

$$\mathcal{L}(\mu) = \underbrace{\mathcal{R}_{\mathcal{B}}(\mathcal{B}) + \mathcal{R}_{\mathcal{T}}(\mathcal{T}) + \mathcal{R}_{\mathcal{N}}(\mathcal{N}; \sigma)}_{\text{Constraint Items}} + \underbrace{\mathcal{F}(\mathcal{B}, \mathcal{T}, \mathcal{N})}_{\text{Fidelity Item}}, \quad (4)$$

where $\mathcal{R}_{\mathcal{B}}(\cdot)$, $\mathcal{R}_{\mathcal{T}}(\cdot)$, and $\mathcal{R}_{\mathcal{N}}(\cdot; \sigma)$ denote the constraint items, and these constraint functions are assumed to be unknown in the latent space. The fidelity item enforces consistency with the latent observation and is usually instantiated as $\mathcal{F}(\mathcal{B}, \mathcal{T}, \mathcal{N}) = \frac{\mu}{2} \|\mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N}\|_F^2$.

Following the Alternating Direction Method of Multipliers (ADMM) [61], we introduce the latent dual variable $\mathcal{Y}$ and write the augmented Lagrangian as:

$$\begin{aligned} \mathcal{L}_{\mu}(\mathcal{B}, \mathcal{T}, \mathcal{N}, \mathcal{Y}) &= \mathcal{R}_{\mathcal{B}}(\mathcal{B}) + \mathcal{R}_{\mathcal{T}}(\mathcal{T}) + \mathcal{R}_{\mathcal{N}}(\mathcal{N}; \sigma) \\ &+ \langle \mathcal{Y}, \mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N} \rangle + \frac{\mu}{2} \|\mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N}\|_F^2. \end{aligned} \quad (5)$$

ADMM decomposes this objective into three main subproblems for $\mathcal{B}$, $\mathcal{T}$, and $\mathcal{N}$. Since these subproblems share the same proximal form, we take the $\mathcal{B}$ -subproblem as the representative case, with the detailed derivation provided in Appendix A:

$$\begin{aligned} \mathcal{B}^k &= \arg \min_{\mathcal{B}} \mathcal{R}_{\mathcal{B}}(\mathcal{B}) \\ &+ \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B} - \mathcal{T}^{k-1} - \mathcal{N}^{k-1} + \frac{1}{\mu} \mathcal{Y}^{k-1} \right\|_F^2. \end{aligned} \quad (6)$$

The next subsection details how LCPNet turns this implicit latent proximal step into a direct learnable solver.

B. Latent Consistent Proximal (LCP) Solver

We derive the LCP solver from the $\mathcal{B}$ -subproblem. For compactness, define the problem as:

$$\mathcal{V}_B^{k-1} = \mathcal{X} - \mathcal{T}^{k-1} - \mathcal{N}^{k-1} + \frac{1}{\mu_B} \mathcal{Y}^{k-1}. \quad (7)$$

Then the $\mathcal{B}$ update can be written in the proximal form

$$\mathcal{B}^k = \arg \min_{\mathcal{B}} \mathcal{R}_{\mathcal{B}}(\mathcal{B}) + \frac{\mu_B}{2} \|\mathcal{B} - \mathcal{V}_B^{k-1}\|_F^2. \quad (8)$$

In conventional optimization-based IRSTD models, the regularizer $\mathcal{R}_{\mathcal{B}}(\cdot)$ is manually specified to encode a prescribed background prior, such as low-rank priors with nuclear norm $\|\cdot\|_*$ constraints. However, there is an inherent gap between manually specified priors and the true physical attributes, so no analytical constraint can fully characterize the desired component. We therefore regard $\mathcal{R}_{\mathcal{B}}(\cdot)$ as an unknown latent regularizer. Specifically, assume that $\nabla \mathcal{R}_{\mathcal{B}}$ is L_B -Lipschitz continuous:

$$\|\nabla \mathcal{R}_{\mathcal{B}}(\mathcal{U}) - \nabla \mathcal{R}_{\mathcal{B}}(\mathcal{V})\|_F \leq L_B \|\mathcal{U} - \mathcal{V}\|_F. \quad (9)$$

Under the Lipschitz condition in Eq. 9, the smooth Taylor upper bound [50] gives:

$$\begin{aligned} \mathcal{R}_{\mathcal{B}}(\mathcal{B}) &\leq \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1}) + \langle \nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1}), \mathcal{B} - \mathcal{B}^{k-1} \rangle \\ &+ \frac{L_B}{2} \|\mathcal{B} - \mathcal{B}^{k-1}\|_F^2, \quad \forall \mathcal{B} \in \mathbb{R}^{H \times W \times C}. \end{aligned} \quad (10)$$

Substituting the majorization in Eq. 10 into the proximal subproblem in Eq. 8, and removing constants independent of $\mathcal{B}$, gives the surrogate objective:

$$\hat{\mathcal{J}}_{\mathcal{B}}(\mathcal{B}) = \frac{L_B}{2} \left\| \mathcal{B} - \left(\mathcal{B}^{k-1} - \frac{1}{L_B} \nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1}) \right) \right\|_F^2 + \frac{\mu_B}{2} \left\| \mathcal{B} - \mathcal{V}_B^{k-1} \right\|_F^2. \quad (11)$$

The surrogate objective in Eq. 11 is the sum of two quadratic terms. Its minimizer admits the following closed-form update, with the detailed derivation provided in Appendix B:

$$\mathcal{B}^k = \frac{L_B}{L_B + \mu_B} \mathcal{B}^{k-1} + \frac{\mu_B}{L_B + \mu_B} \mathcal{V}_B^{k-1} - \frac{\nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1})}{L_B + \mu_B}. \quad (12)$$

The closed-form solution in Eq. 12 is strictly equivalent to a gradient-like descent step:

$$\mathcal{B}^k = \mathcal{B}^{k-1} - \frac{1}{L_B + \mu_B} \left(\nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1}) + \mu_B (\mathcal{B}^{k-1} - \mathcal{V}_B^{k-1}) \right). \quad (13)$$

Equivalently, this update can be written as:

$$\mathcal{B}^k = \mathcal{B}^{k-1} - \eta_B \mathcal{G}_B^{k-1}, \quad (14)$$

where $\eta_B = 1/(L_B + \mu_B)$. This formulation explicitly anchors the update on the previous state $\mathcal{B}^{k-1}$, providing a more directly reference for estimating the next background state $\mathcal{B}^k$. By substituting Eq. 7 into Eq. 13 and defining the current latent residual as $\mathcal{P}^{k-1} = \mathcal{X} - \mathcal{B}^{k-1} - \mathcal{T}^{k-1} - \mathcal{N}^{k-1}$, the ideal update direction is directly obtained as:

$$\mathcal{G}_B^{k-1} = \nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1}) - \mu_B \mathcal{P}^{k-1} - \mathcal{Y}^{k-1}. \quad (15)$$

However, directly computing $\mathcal{G}_B^{k-1}$ inside a network module is not practical. First, $\mathcal{R}_{\mathcal{B}}(\cdot)$ is an unknown latent regularizer, so $\nabla \mathcal{R}_{\mathcal{B}}(\mathcal{B}^{k-1})$ has no explicit analytical form. Second, explicitly concatenating $\mathcal{Y}^{k-1}$, $\mathcal{B}^{k-1}$, and $\mathcal{P}^{k-1}$ as solver inputs increases the input complexity of each unfolding stage. Third, the true update direction should not depend only on the current variables, it should also reflect the historical optimization trajectory across previous stages. Therefore, we replace the ideal direction in Eq. 15 with a learnable latent proximal surrogate:

$$\mathcal{G}_B^{k-1} \approx \tilde{\mathcal{G}}_B^{k-1} = \Psi_B(\mathcal{B}^{k-1} - \mathcal{P}^{k-1}, h^k). \quad (16)$$

Instead of directly concatenating all solver variables, $\mathcal{B}^{k-1} - \mathcal{P}^{k-1}$ serves as a compact variable-centered residual that fuses the current background estimate and the decomposition-consistency violation into a single surrogate cue. Based on this approximation, we obtain a more direct Markov-like LCP solver that incorporates historical trajectories:

$$\mathcal{B}^k = \mathcal{B}^{k-1} - \eta_B \Psi_B(\mathcal{B}^{k-1} - \mathcal{P}^{k-1}, h^k), \quad (17)$$

h^k complements the current variables with global memory from previous stages. The construction of this shared state and its role in encoding historical optimization trajectories are detailed in Section III-C.

C. Shared Optimization Memory (SOM)

Existing recurrent unfolding methods usually place the memory mechanism inside a specific branch of the unfolding network. For example, RPCANet++ introduces a memory-augmented module to alleviate inter-stage transmission loss in background approximation module, mainly aiming to preserve background features across stages [18]. The stored memory mainly serves the background branch, and thus cannot directly modulate the target and noise updates. The target branch can only benefit from this memory indirectly through the background representation. Moreover, the variables $\mathcal{B}$, $\mathcal{T}$, $\mathcal{N}$, $\mathcal{Y}$, and $\mathcal{X}$ are not encoded as a coupled decomposition system, so the historical optimization trajectory is only partially represented.

To address these limitations, LCPNet constructs a SOM for the whole decomposition process. At stage k , we form a joint observation by concatenating the current latent variables:

$$o_k = [\mathcal{B}^k, \mathcal{T}^k, \mathcal{N}^k, \mathcal{Y}^k, \mathcal{X}]. \quad (18)$$

An optimization state cell then maps this joint observation into a shared state h_k through a gated recurrent mechanism [52]:

$$\begin{aligned} z_k &= \sigma(W_z * [o_k, h_{k-1}] + b_z), \\ r_k &= \sigma(W_r * [o_k, h_{k-1}] + b_r), \\ \tilde{h}_k &= \tanh(W_q * [o_k, r_k \odot h_{k-1}] + b_q), \\ h_k &= (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k. \end{aligned} \quad (19)$$

As shown in Fig 5(c), $*$ denotes convolution, z_k is the update gate, r_k is the reset gate, and $\odot$ denotes element-wise multiplication. Unlike branch-specific memory, h_k is shared by the $\mathcal{B}$, $\mathcal{T}$, and $\mathcal{N}$ update branches and is directly fed into their proximal surrogate updaters. Therefore, the same system-level state can modulate all primal updates, making h_k a global control variable for the unfolding solver.

The adaptive gates aggregate historical decomposition observations into a high-order effective memory of the unfolding trajectory. Recursively expanding Eq. 19 gives

$$h_k = \left(\prod_{i=1}^k (1 - z_i) \right) \odot h_0 + \sum_{t=1}^k \left[z_t \odot \tilde{h}_t \odot \prod_{i=t+1}^k (1 - z_i) \right]. \quad (20)$$

Eq. 20 shows that h_k is a gated accumulation of all previous candidate states, where historical information is adaptively retained across stages. The detailed proof is provided in Appendix C, and this expansion indicates that SOM forms a shared high-order memory for the whole latent system rather than isolated branch-wise states.

D. Network Architecture

The overall architecture of LCPNet is shown in Fig. 4. Given an infrared image, LCPNet first lifts the input into latent domain and then performs K stages of latent decomposition unfolding. As illustrated in Fig. 5(a), each stage updates the background, target, noise, and dual variables under the latent decomposition constraint, while SOM provides historical guidance for the stage-wise proximal surrogate updates. The final

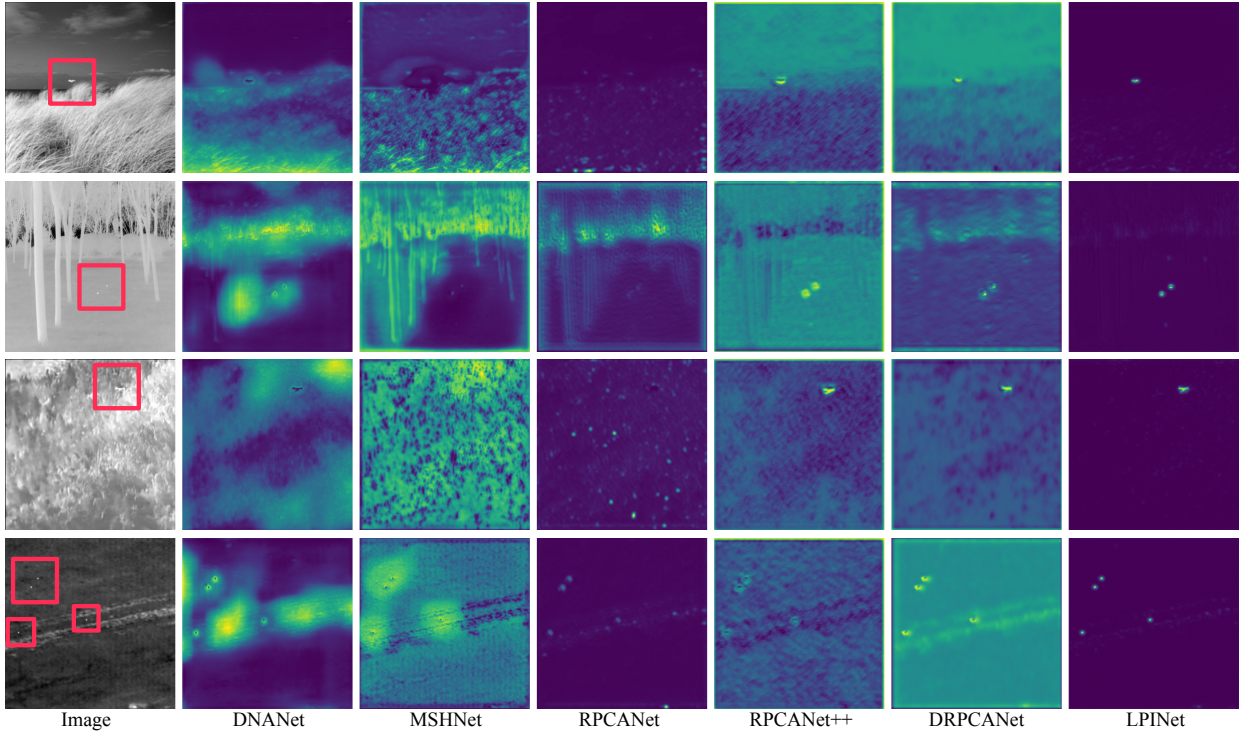


Fig. 6. Visualization of the last layer feature maps produced by different methods. For each method, the feature map is projected by PCA, and the most informative principal component is selected for visualization. Rows 1, 3, 4 are 001322, 000562, 000377 in NUDT-SIRST [9]. Row 2 is XD902 in IRSTD-1k [7].

target representation is decoded back to the image domain to generate the detection map.

The update module in each stage is shown in Fig. 5(b). Following the LCP solver derived in Eq. 16, the module estimates the learned proximal direction from the variable residual and shared optimization state. The same design is used for different primal variable branches.

For the updater in LCP solver, a standard batch normalization design is sensitive to mini-batch statistics, which is not ideal for IRSTD because infrared backgrounds vary across scenes and small targets occupy only a few pixels [19]. We therefore use Group Normalization (GN) to normalize features within each sample, reducing the dependence on batch-level statistics while preserving scene-specific feature variations [20]. Furthermore, Spectral Normalization (SN) is adopted to constrain the gain of the convolutional updater, preventing background fluctuations from being repeatedly amplified and improving the discrimination of weak target responses:

$$\text{SN}(\mathbf{W}) = \frac{\mathbf{W}}{\sigma_{\max}(\mathbf{W})}, \quad (21)$$

where $\sigma_{\max}(\mathbf{W})$ denotes the largest singular value of the convolutional weight $\mathbf{W}$, and the normalized operator is used in the recurrent unfolding updater [22]. As shown in Fig. 5(b), the final update module combines this gain-controlled convolution with sample-wise normalization and nonlinear activation, providing a smoother approximation to the learned proximal direction $\tilde{\mathcal{G}}_B^{k-1}$.

IV. EXPERIMENTS

This section evaluates the proposed method from several complementary perspectives. We first introduce the datasets, metrics, and implementation settings, and then compare the proposed method with SOTAs in terms of quantitative accuracy, visual detection quality, and computational efficiency. Then, ablation studies are conducted to examine the contribution of each key design. Finally, we discuss the differences and limitations of our method compared to other approaches, as well as our outlook for the future.

A. Implementation Details

1) *Datasets and Metrics*: We evaluate LCPNet on four publicly available benchmarks, namely NUDT-SIRST [9], IRSTD-1K [7], SIRST [8], and SIRST-Aug [11]. NUDT-SIRST [9] contains 1327 images with a resolution of 256×256 , and is widely used to evaluate robustness under diverse backgrounds and complex clutter conditions. Following the protocol of [9], the training and test sets are split 1:1. IRSTD-1K [7] is a realistic infrared small target detection benchmark, comprising manually annotated real-world infrared images. It contains 800 training images and 201 test images. SIRST [8] is one of the earliest and most widely used single-frame benchmarks. It contains 341 training images and 86 test images in various sizes. SIRST-Aug [11] is an augmented version derived from SIRST [8] through data augmentation. It contains 8525 training images and 545 test images and provides more sufficient training samples for deep models.

TABLE I

COMPARISON WITH STATE-OF-THE-ART IRSTD METHODS IN TERMS OF IOU(%), F_1 (%), P_d (%) AND F_a (10^{-5}) ON FOUR BENCHMARKS. THE BEST AND SECOND-BEST RESULTS UNDER EACH METRIC ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY. INFERENCE SPEED IS MEASURED ON AN NVIDIA GeForce RTX 3090 GPU AND AN AMD EPYC 7502 CPU.

Model	NUDT-SIRST				IRSTD-1K				SIRST				SIRST-Aug				Params		Flops	Time(ms)	
	IoU↑	F ₁ ↑	P _d ↑	F _a ↓	IoU↑	F ₁ ↑	P _d ↑	F _a ↓	IoU↑	F ₁ ↑	P _d ↑	F _a ↓	IoU↑	F ₁ ↑	P _d ↑	F _a ↓	(M)	(G)	(CPU/GPU)		
HVS-Based Methods																					
Tophat [25]	22.25	36.41	91.21	75.73	6.64	12.46	81.09	14.24	26.93	42.44	88.99	18.27	16.37	28.14	81.15	160.3	-	-	0.08/-		
Max-Median [26]	20.49	34.01	84.97	31.27	20.76	34.38	85.56	21.21	32.56	49.12	94.46	3.38	16.88	28.89	79.36	8.25	-	-	7836/-		
MPCM [27]	9.26	16.95	70.58	32.72	15.70	27.13	62.54	12.47	22.30	36.47	90.83	5.56	19.76	33.00	93.39	3.14	-	-	69.78/-		
HBMLCM [24]	7.60	14.13	60.74	17.84	12.60	22.38	57.73	10.08	11.95	21.36	79.81	41.32	2.37	46.37	50.61	146.1	-	-	6107/-		
Optimization-Based Methods																					
IPI [5]	34.62	51.43	92.38	7.54	25.27	40.35	83.51	20.93	44.83	61.90	97.25	3.76	21.93	35.97	80.05	3.62	-	-	8004/-		
NRAM [14]	11.44	20.52	65.71	2.35	17.97	30.46	59.79	5.71	26.39	41.75	88.07	1.60	10.41	18.86	76.06	3.26	-	-	2443/-		
NOLC [62]	18.60	31.37	70.68	3.96	10.90	19.65	64.26	1.83	20.35	33.82	83.48	1.36	8.68	15.97	59.00	6.69	-	-	102.3/-		
PSTNN [15]	25.17	40.22	80.21	7.61	24.39	39.22	65.29	15.77	35.17	52.04	88.99	2.57	12.38	22.04	60.79	2.98	-	-	62.46/-		
Deep learning-Based Methods																					
ACM [8]	69.46	81.98	97.14	13.11	54.16	70.25	83.84	7.32	66.77	80.07	98.16	11.03	67.79	80.80	95.87	32.52	0.51	0.42	-7.17		
AGPCNet [11]	85.02	91.90	97.88	4.34	60.44	75.34	90.72	6.10	68.19	81.08	99.10	12.09	72.07	83.77	98.62	29.39	12.4	43.0	-14.71		
DNANet [9]	87.73	93.47	97.67	5.52	62.23	76.72	92.44	5.64	73.12	84.48	99.08	10.26	70.56	82.74	96.56	36.49	4.69	13.9	-22.64		
UIUNet [10]	88.94	94.15	95.23	1.53	63.08	77.36	92.10	6.09	72.15	83.82	98.16	7.90	70.76	82.88	96.70	34.30	50.5	54.2	-27.50		
MSHNet [12]	75.96	86.34	95.23	8.28	64.83	78.67	92.78	7.14	64.64	78.52	98.16	15.66	71.93	83.67	99.03	32.06	4.06	6.01	-13.52		
SCTransNet [63]	89.49	94.45	97.67	1.02	62.04	76.56	73.47	2.36	71.15	83.14	97.24	0.89	72.05	83.20	97.76	5.20	11.3	10.0	-31.42		
ALCNet [6]	80.75	89.35	97.56	9.56	60.69	75.53	85.57	5.17	70.31	82.57	99.08	11.36	67.38	80.51	97.94	28.68	0.43	0.38	-9.51		
ISNet [7]	87.05	93.08	96.83	4.05	63.00	77.29	88.66	5.52	66.96	80.20	96.33	9.85	71.10	83.11	97.66	30.33	1.08	30.3	-28.30		
IRSAM [28]	87.85	93.53	99.05	3.21	63.58	77.74	92.10	4.21	67.97	80.93	99.08	8.34	72.29	83.95	99.02	25.94	10.5	37.6	-75.06		
L2SKNet [29]	93.08	96.42	97.88	1.35	57.07	72.67	90.03	37.90	62.54	76.96	97.25	10.63	71.89	83.65	98.62	31.56	1.07	5.92	-9.38		
Deep Unfolding-Based Methods																					
RPCANet [17]	89.31	94.35	97.14	2.87	63.21	77.45	88.31	4.39	70.37	82.61	95.41	7.42	72.54	84.08	98.21	34.14	0.68	44.3	-15.40		
RPCANet++ [18]	94.39	97.12	98.41	1.34	64.93	78.73	89.70	4.35	74.63	85.47	100.0	10.22	69.97	82.33	98.76	28.00	2.96	190	-43.61		
DRPCANet [64]	94.16	96.99	98.41	1.35	64.14	78.15	92.09	5.16	75.52	86.05	99.08	7.15	76.50	86.69	98.62	30.65	1.16	73.2	-86.50		
LCPNet-4	95.70	97.80	97.57	0.03	65.15	78.89	86.60	1.58	71.18	83.17	94.50	0.87	72.55	84.09	95.74	2.87	1.15	75.2	-23.54		
LCPNet-6	95.77	97.84	97.35	0.09	65.95	79.48	87.63	1.55	72.95	84.36	96.33	0.89	72.49	84.05	96.84	2.29	1.72	112	-31.38		

For quantitative evaluation, we adopt five commonly used metrics, including mean Intersection over Union (IoU), F_1 score, Probability of detection (P_d), and F_a false alarm rate (F_a). They assess performance from the perspectives of overall segmentation quality, the ability to correctly detect real objects, and the risk of false alarms in cluttered backgrounds, respectively. Additionally, we included area under the ROC curve (AUC) to summarize the detection performance at different decision thresholds based on the ROC curve.

2) *Experimental Settings*: All implementations were developed based on PyTorch. The experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU and an AMD EPYC 7502 CPU. For training, all input images were resized to a base size of 256×256 , and the batch size was set to 8. The model was trained for 800 epochs on NUDT-SIRST, IRSTD-1K, and SIRST, and for 400 epochs on SIRST-Aug. We used the Adam optimizer with an initial learning rate of 1×10^{-4} , and adopted the poly learning rate

scheduler with a power of 0.9. The network is supervised by SoftIoU loss [3].

B. State-of-the-Art Comparison

To provide a comprehensive evaluation, we compare LCPNet with representative IRSTD methods on NUDT-SIRST [9], IRSTD-1K [7], SIRST [8], and SIRST-Aug [11] under the same evaluation protocol described above. The compared methods cover four categories: HVS-based methods, optimization-based methods, deep learning networks, and recent deep unfolding models. This selection allows us to examine detection accuracy, false-alarm suppression, visual robustness, and the efficiency-accuracy tradeoff across different model families.

1) *Quantitative Comparison*: Table I shows that LCPNet provides the most consistent quantitative performance among the compared model families. On the two larger benchmarks e.g. NUDT-SIRST [9], IRSTD-1K [7], LCPNet-6 obtains

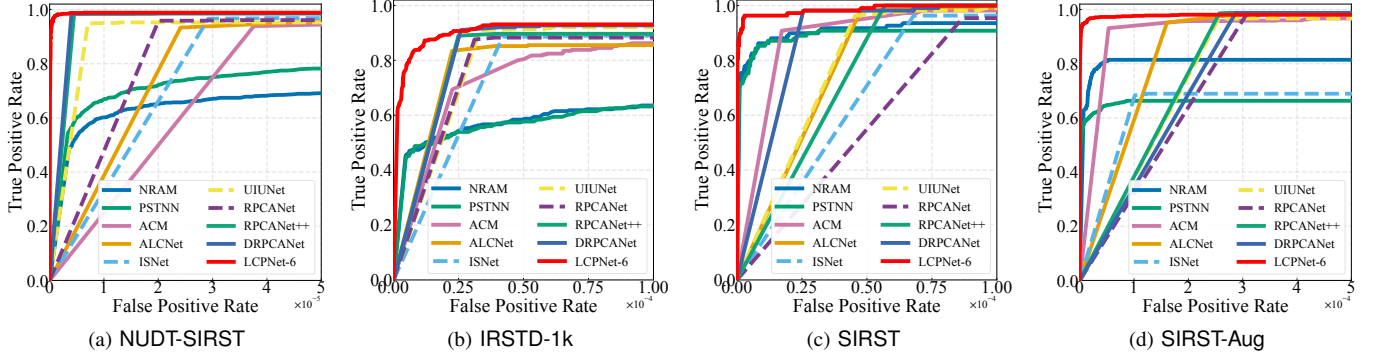


Fig. 7. ROC curves of different state-of-the-art methods across four infrared small target detection datasets. Our method is represented by the red line.

the best performance, with IoU/ F_1 of 95.77%/97.84% and 65.95%/79.48%, respectively. More importantly for IRSTD, where small bright clutter easily causes false alarms, the LCPNet variants achieve the lowest F_a on all four benchmarks, indicating that the proposed decomposition is not merely more accurate in overlap but also more conservative in suppressing background responses.

This behavior is closely related to the design of LCPNet. The latent-space unfolding gives each stage a higher-dimensional representation for separating target and background structures, while the proximal surrogate and shared memory stabilize the update trajectory across stages. As a result, LCPNet is particularly strong in cluttered scenes. Even when some baselines obtain higher individual P_d or IoU/ F_1 on SIRST and SIRST-Aug, LCPNet reduces F_a to 0.87 and 2.29, respectively, showing a better balance between detecting weak targets and avoiding spurious responses.

The ROC results in Fig. 7 provide the same evidence from a threshold-independent view: after zooming into the high-sensitivity region, the red LCPNet curve stays closest to the upper-left corner. Consistently, Table II reports the best AUC on three benchmarks, where LCPNet-6 reaches 0.9941, 0.9656, and 0.999998, and remains competitive on the augmented one with an AUC of 0.9803. Overall, these results suggest that the proposed latent space unfolding strategy improves not only peak segmentation metrics but also the robustness of detection decisions across thresholds.

2) *Qualitative Visual Comparison:* Fig. 8-Fig. 11 provide representative visual comparisons, where green boxes denote correct detections and red boxes denote false alarms or missed targets. These examples cover several typical difficult cases, including tree-branch clutter, multiple adjacent targets, cloud background structures, and complex terrain textures.

In Fig. 8, traditional optimization-based methods such as MPCM, NRAM, and PSTNN leave many clutter responses, while only RPCANet and LCPNet-6 correctly detect the true target. This result suggests that unfolding-based decomposition is more robust than hand-crafted priors when the background contains structured bright interference. The advantage of LCPNet becomes clearer in Fig. 9 with multiple targets. Most competing methods either miss one target or mix the target response with nearby background, whereas LCPNet-6 completely detects both targets. For the cloud and

TABLE II
AUC COMPARISON OF DIFFERENT METHODS ON FOUR INFRARED SMALL TARGET DETECTION DATASETS. HIGHER VALUES ARE MARKED AS **BOLD** AND INDICATE BETTER DETECTION PERFORMANCE.

Method	NUDT-SIRST	IRSTD-1K	SIRST	SIRST-Aug
NRAM [14]	0.7403	0.7777	0.936405	0.7794
PSTNN [15]	0.8968	0.8114	0.892235	0.5069
ACM [8]	0.9867	0.9363	0.990813	0.9834
ALCNet [6]	0.9909	0.9281	0.999975	0.9909
ISNet [7]	0.9861	0.9446	0.981616	0.8436
UIUNet [10]	0.9761	0.9604	0.995389	0.9833
RPCANet [17]	0.9804	0.9415	0.961126	0.9909
RPCANet++ [18]	0.9925	0.9484	0.999972	0.9937
DRPCANet [64]	0.9931	0.9639	0.999986	0.9936
LCPNet-6	0.9941	0.9656	0.999998	0.9803

terrain examples in Fig. 10 and Fig. 11, many baselines are distracted by background textures and generate multiple red-box responses. LCPNet-6 localizes the real targets with more compact responses and fewer clutter activations, although a small residual false alarm remains in the cloud scene. Overall, the visual results support the quantitative findings: LCPNet improves practical detection reliability by better separating small targets from structured background clutter.

To further inspect the feature-level behavior, Fig. 6 visualizes the last-layer feature maps of different methods using the most informative PCA component. Across sky and ground background scenes, as well as single-target, multi-target, and heavy-clutter cases, LCPNet produces clearer target-focused activations than the competing methods. The target responses are more concentrated around true small objects, while background structures are less likely to dominate the heatmaps. This suggests that the proposed latent space unfolding process provides more reliable feature evidence for subsequent target detection and segmentation.

3) *Computational Efficiency Analysis:* Beyond detection accuracy, Table I also reports the model parameters, FLOPs, and CPU/GPU inference time. HVS-based methods do not involve trainable parameters, but their CPU runtime varies greatly because they depend on hand-crafted filtering or local contrast operations. Optimization-based methods are also parameter-free in the neural-network sense, yet their iterative solvers are generally slow on CPU and lack efficient GPU

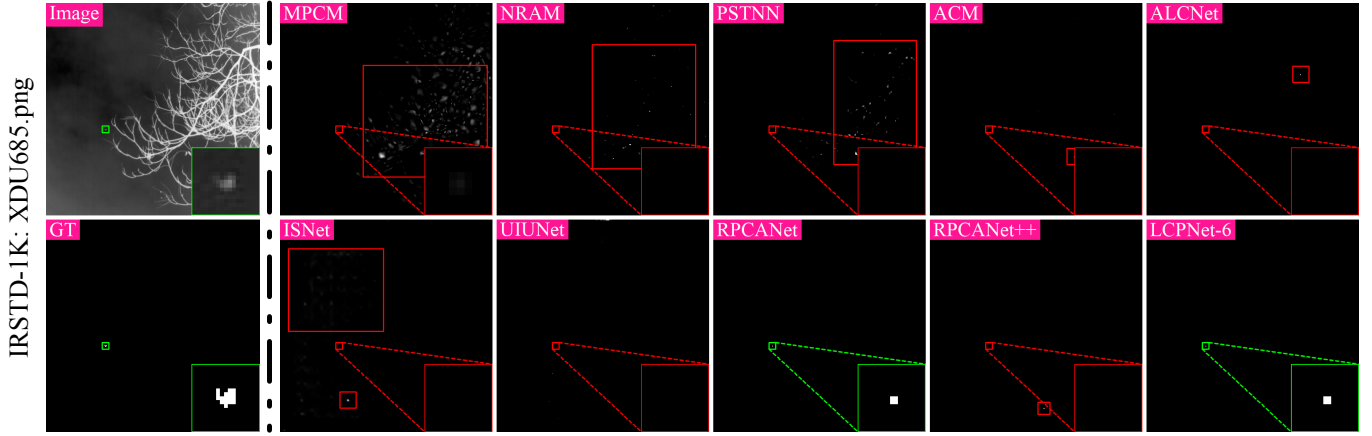


Fig. 8. Qualitative comparison on image *XDU685*. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

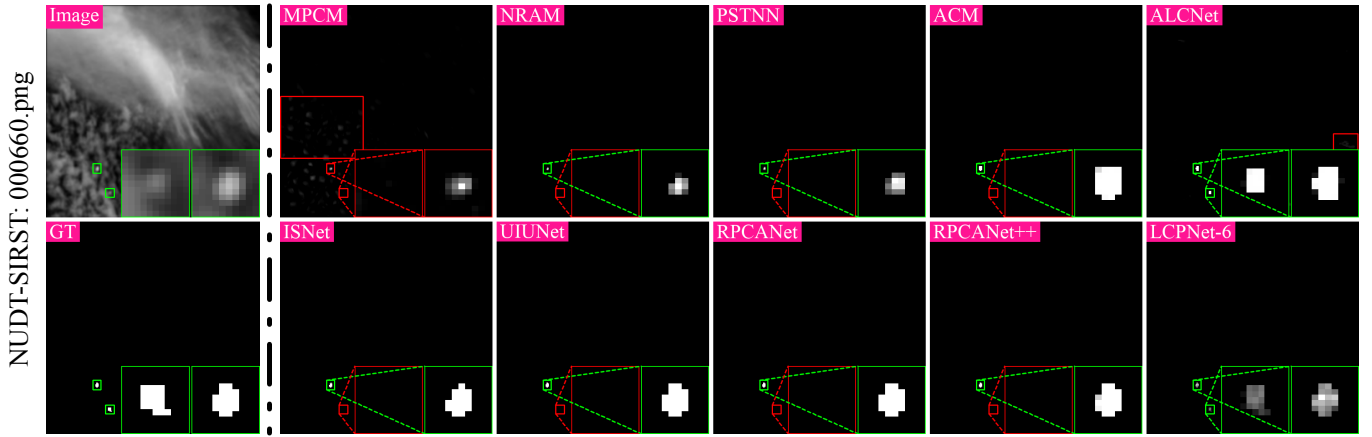


Fig. 9. Qualitative comparison on image *000660*. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

inference.

Deep learning-based methods improve inference efficiency by replacing iterative solvers with feed-forward networks, but their cost differs substantially across architectures. Lightweight models such as ACM and ALCNet are fast, whereas larger context aggregation or nested U-shaped designs introduce noticeably higher FLOPs and latency. However, these faster non-unfolding networks usually trade interpretability and often show weaker robustness in the quantitative and visual comparisons above.

Among deep unfolding-based methods, LCPNet provides a practical balance between accuracy, model size, and runtime. Compared with RPCANet++, LCPNet-4 uses fewer parameters and substantially lower FLOPs, while also running faster on GPU. LCPNet-6 increases the unfolding depth to improve accuracy but still keeps the runtime below heavier unfolding baselines such as RPCANet++ and DRPCANet. Therefore, LCPNet does not achieve its performance by simply enlarging the network. Its efficiency mainly comes from performing compact latent-domain updates with shared optimization memory, yielding a favorable accuracy-efficiency tradeoff for infrared small target detection.

C. Ablation Studies

To verify the effectiveness of each component in LCPNet, we conduct a series of controlled ablation studies by modifying one design factor at a time while keeping the remaining settings unchanged. Together, these studies provide an overall validation of the main architectural choices and their impact on detection performance.

1) *Effect of Execution Domain*: We first examine whether the unfolding process should be executed in the image domain or in the latent domain. In Table III, variants (b) and (c) use the same surrogate-style solver and batch-based updater, so their comparison isolates the effect of the execution domain. Moving the updates from $\mathbf{X}$ to $\mathcal{X}$ improves IoU by 0.72, F_1 by 0.39, and P_d by 0.84, indicating that the latent variables provide a more informative representation for preserving weak target responses. However, the F_a value is not reduced in this setting, which suggests that latent execution alone is insufficient when the updater is weakly constrained.

The benefit becomes clearer when the updater is equipped with stronger regularization. Under the same solver and regularized updater setting, variants (e) and (f) show that latent-domain execution brings gains of 2.83 in IoU, 1.49 in F_1 , and 1.06 in P_d , while reducing F_a from 0.080 to 0.025. This

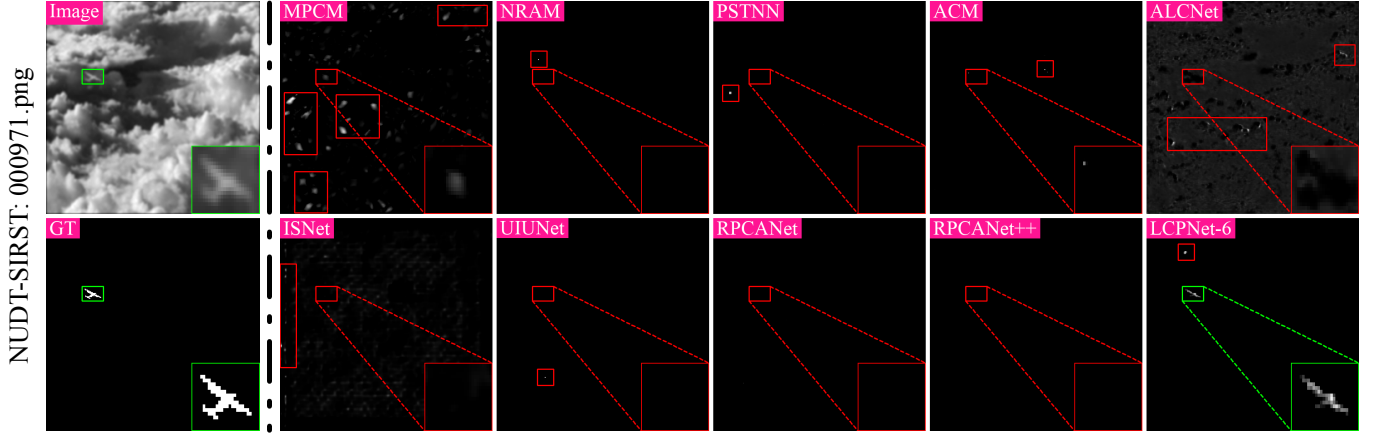


Fig. 10. Qualitative comparison on image 000971. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

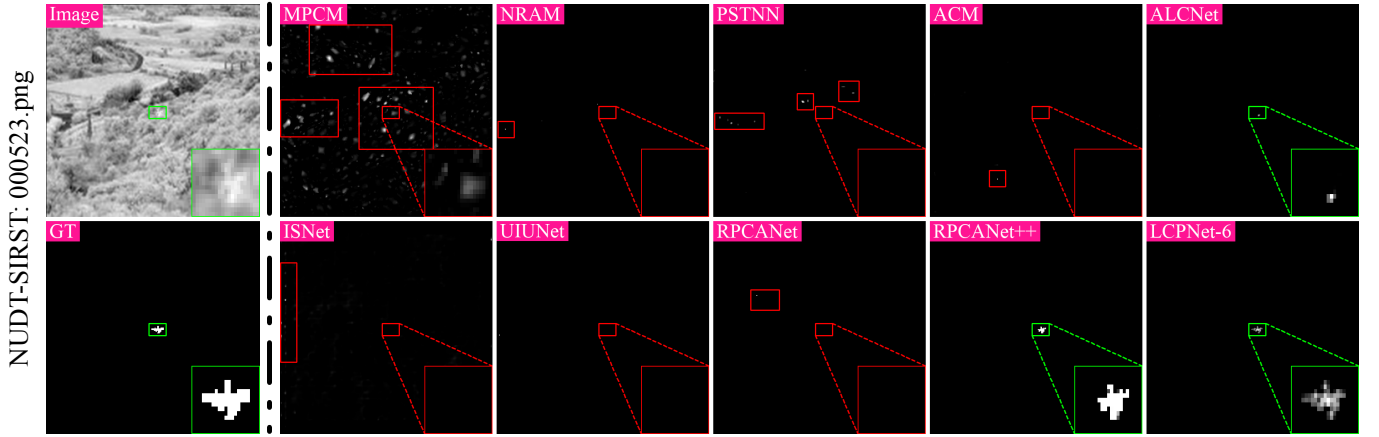


Fig. 11. Qualitative comparison on image 000523. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

result indicates that the expanded latent representation can better separate target and background components when the update process is stable enough to exploit it. Overall, these two comparisons show that latent-domain unfolding is more effective than directly updating image-domain variables.

2) *Effect of Solver Design*: We evaluate the solver design used to generate the stage-wise update direction. The R-style design, adopted by the RPCANet series [17], [18], [64], directly reconstructs the next background variable from the current decomposition residual through a learned correction:

$$\mathbf{B}^k = \mathbf{D}^{k-1} - \mathbf{T}^{k-1} + \mathcal{F}(\mathbf{D}^{k-1} - \mathbf{T}^{k-1}). \quad (22)$$

This form is simple and effective, but the update of $\mathbf{B}^k$ does not explicitly use $\mathbf{B}^{k-1}$. Therefore, the background trajectory is mainly regenerated from the residual at each stage, which weakens the continuity of the unfolding process.

In contrast, the proposed LCP solver design in Eq. 17 keeps the proximal surrogate form while removing the SOM-related state input for a fair comparison. This design updates current latent variable from its previous state and a decomposition-consistency residual, making the learned direction closer to a proximal descent step rather than an indirect residual mapping.

The comparison between variants (a) and (b) in Table III verifies this effect under the same image-domain and batch-based updater setting. Replacing the R-style solver with the S-style solver brings gains of 1.27 in IoU and 0.71 in F_1 , while lowering F_a by 2.72. Although P_d decreases by 3.38, the much lower false alarm rate and higher overlap metrics indicate that the S-style solver suppresses background responses more effectively. Overall, the proximal surrogate formulation provides a more stable and discriminative update rule for the unfolding decomposition.

3) *Effect of Updater Regularization*: We further analyze the regularization strategy inside the updater. In Table III, variants (c), (d), and (f) keep the latent execution domain and S-style solver unchanged, and only vary the updater regularization.

Replacing BN with GN gives the main improvement. From variant (c) to (d), IoU, F_1 , and P_d increase by 4.29, 2.29, and 2.65, respectively, while F_a is reduced by 0.260. This large gain is consistent with the characteristics of IRSTD, where infrared scenes often have heterogeneous background statistics and small targets occupy only a tiny portion. A batch-dependent updater can therefore be sensitive to mixed scene statistics, whereas GN normalizes features within each sample and provides a more stable update direction.

TABLE III

ABLATION STUDY OF EXECUTION DOMAIN, SOLVER DESIGN, AND UPDATER REGULARIZATION IN TERMS OF IoU(%), F_1 (%), P_d (%) AND $F_a(10^{-5})$ ON NUDT-SIRST [9]. $\mathbb{B}$, $\mathbb{G}$ AND $\mathbb{S}$ REPRESENT BATCH NORMALIZATION, GROUP NORMALIZATION, AND SPECTRAL NORMALIZATION, RESPECTIVELY.

Index	Domain	Solver Design	Updater Regular	NUDT-SIRST			
				IoU $\uparrow$	$F_1\uparrow$	$P_d\uparrow$	$F_a\downarrow$
(a)	Image	R-style	$\mathbb{B}$	89.31	94.35	97.14	2.87
(b)	Image	LCP	$\mathbb{B}$	90.58	95.06	93.76	0.152
(c)	Latent	LCP	$\mathbb{B}$	91.30	95.45	94.60	0.389
(d)	Latent	LCP	$\mathbb{G}$	95.59	97.74	97.25	0.129
(e)	Image	LCP	$\mathbb{G} + \mathbb{S}$	92.87	96.31	96.51	0.080
(f)	Latent	LCP	$\mathbb{G} + \mathbb{S}$	95.70	97.80	97.57	0.025

TABLE IV

ABLATION STUDY OF MEMORY TYPE IN TERMS OF IoU(%), F_1 (%), P_d (%) AND $F_a(10^{-5})$ ON NUDT-SIRST [9] AND IRSTD-1K [7].

Memory	NUDT-SIRST				IRSTD-1K			
	IoU $\uparrow$	$F_1\uparrow$	$P_d\uparrow$	$F_a\downarrow$	IoU $\uparrow$	$F_1\uparrow$	$P_d\uparrow$	$F_a\downarrow$
None	95.12	97.50	96.51	0.074	64.35	78.31	86.60	1.45
Cat	95.33	97.61	96.40	0.051	64.31	78.28	85.57	1.41
ConvLSTM	95.43	97.66	97.25	0.046	64.72	78.58	86.94	1.38
SOM	95.70	97.80	97.57	0.025	65.15	78.89	86.60	1.58

Adding spectral normalization on top of GN brings a smaller but consistent gain. From variant (d) to (f), IoU and F_1 further increase by 0.11 and 0.06, P_d improves by 0.32, and F_a is further reduced by 0.104. Overall, GN is the primary factor for improving updater stability, and SN further regularizes the unfolding update process, leading to more reliable target-background separation.

4) *Effect of Memory Type*: We then evaluate how historical information is represented across unfolding stages. Table IV compares four settings, including no memory, simple concatenation, ConvLSTM-style recurrent memory used in RPCANet++ [18], and the proposed SOM. Compared with no memory, simple concatenation brings only a marginal IoU gain of 0.21 on NUDT-SIRST and shows no clear improvement on IRSTD-1K, indicating that static feature aggregation is insufficient for modeling the stage-wise optimization history. ConvLSTM gives more consistent gains, improving IoU by 0.31 and 0.37 on the two benchmarks and reducing F_a on both datasets. This result confirms that recurrent memory is useful for preserving cross-stage information.

SOM further improves segmentation quality by treating the decomposition variables as a coupled optimization state rather than storing branch-specific memory. Compared with ConvLSTM, SOM increases IoU/ F_1 by 0.27/0.14 on NUDT-SIRST and 0.43/0.31 on IRSTD-1K. It also improves P_d on NUDT-SIRST by 0.32 and reduces F_a by 0.021, although its F_a on IRSTD-1K is not the lowest. Overall, SOM provides stronger overlap accuracy and more stable target preservation, showing that a shared optimization memory is more effective than others.

TABLE V

ABLATION STUDY OF THE UNFOLDING STAGE NUMBER IN TERMS OF IoU(%), F_1 (%), P_d (%) AND $F_a(10^{-5})$ ON NUDT-SIRST [9] AND IRSTD-1K [7].

Stages	NUDT-SIRST				IRSTD-1K			
	IoU $\uparrow$	$F_1\uparrow$	$P_d\uparrow$	$F_a\downarrow$	IoU $\uparrow$	$F_1\uparrow$	$P_d\uparrow$	$F_a\downarrow$
K=2	94.27	97.05	95.03	0.046	64.28	78.25	82.47	1.12
K=3	95.16	97.52	97.14	0.011	64.36	78.31	84.88	1.29
K=4	95.70	97.80	97.57	0.025	65.15	78.89	86.60	1.58
K=5	96.11	98.02	98.52	0.028	65.85	79.41	87.63	1.53
K=6	95.77	97.84	97.35	0.092	65.95	79.48	87.63	1.55

5) *Effect of Stage Number*: Finally, we study the influence of the unfolding stage number in Table V. Increasing the number of stages generally improves the detection quality because more decomposition updates allow the latent variables and the shared state to be refined progressively. For example, increasing the depth from $K = 2$ to $K = 4$ improves IoU/ F_1 by 1.43/0.75 on NUDT-SIRST and 0.87/0.64 on IRSTD-1K, while P_d also increases on both benchmarks.

The improvement becomes less regular when the network is further deepened. The setting $K = 5$ gives a notably high result on NUDT-SIRST, but this behavior is more like an isolated fluctuation under the current evaluation rather than the main trend. We therefore do not take it as the primary result. Overall, a moderate-to-deep unfolding depth provides better representation refinement, while excessively increasing the number of stages brings smaller and less stable gains.

D. Discussion and Analysis

1) *Technical Discussion*: LCPNet follows the same model-driven spirit as recent deep unfolding methods, but differs from them in how the decomposition process is represented and updated. Traditional optimization-based methods operate directly in the image domain and rely on hand-crafted low-rank/sparse priors, while RPCANet-style networks unfold an RPCA-like process with learnable modules in the image domain. LCPNet instead lifts the observation and decomposition variables into a latent domain. This transformation does not discard the physical decomposition constraint, but gives each stage a higher-dimensional interface where small target responses, background structures, and noise components can evolve more flexibly.

The solver design also differs from the residual-style update used in RPCANet-related methods. In the R-style formulation, the next background estimate is mainly regenerated from the current residual through a learned correction, and the previous background state is not explicitly involved in the update. LCPNet adopts an S-style proximal surrogate update, where each variable is updated from its previous state and the current decomposition-consistency residual. This makes the learned update closer to a proximal descent step and gives the unfolding trajectory stronger continuity. Moreover, the SOM further separates LCPNet from RPCANet++. The memory module in RPCANet++ mainly serves a specific branch, whereas SOM

encodes the coupled state of $\mathcal{B}$, $\mathcal{T}$, $\mathcal{N}$, $\mathcal{Y}$, and $\mathcal{X}$ and feeds the same historical state to all primal-variable updates. Through the gated recurrence, this state forms an unbounded-order effective memory of the whole latent decomposition system rather than a branch-level feature cache.

2) *Limitations and Future Work*: Although LCPNet achieves a favorable accuracy-efficiency tradeoff, its inference time still has room for further reduction. The latent-domain updates and recurrent state propagation introduce additional computation compared with very lightweight feed-forward detectors, especially when more unfolding stages are used. Future work can further compress the updater, simplify the state transition, or design adaptive early-exit strategies so that easy samples require fewer stages. We expect these directions to support more efficient and robust infrared small target detectors, especially for real-time deployment on resource-constrained platforms.

V. CONCLUSION

In this paper, we introduced LCPNet for infrared small target detection. By confirming that low-rank decomposition prior remains valid in latent representations, LCPNet extends unfolding from image domain to latent domain, preserving the physical constraint while avoiding repeated compression of intermediate states. Through modeling and detailed derivations on latents, the proposed LCP solver turns indirect residual reconstruction into direct proximal-state evolution and introduces regularized update dynamics for stable small target separation. The SOM further upgrades branch-specific memory into a shared optimization state, so historical information can guide all coupled decomposition variables across stages. Extensive experiments on four public benchmarks demonstrate the effectiveness of LCPNet. Quantitative and qualitative results show that LCPNet achieves accurate and robust detection across cluttered scenes and multiple-target cases. Ablation studies progressively demonstrate that these benefits stem from the proposed improvements. Overall, LCPNet provides an effective and interpretable framework forIRSTD.

REFERENCES

- [1] M. Zhao, W. Li, L. Li, J. Hu, P. Ma, and R. Tao, "Single-frame infrared small-target detection: A survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 87–119, 2022.
- [2] R. Kou, C. Wang, Z. Peng, Z. Zhao, Y. Chen, J. Han, F. Huang, Y. Yu, and Q. Fu, "Infrared small target segmentation networks: A survey," *Pattern Recognition*, vol. 143, p. 109788, 2023.
- [3] H. Wang, L. Zhou, and L. Wang, "Miss detection vs. false alarm: Adversarial learning for small object segmentation in infrared images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8509–8518.
- [4] C. P. Chen, H. Li, Y. Wei, T. Xia, and Y. Y. Tang, "A local contrast method for small infrared target detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 574–581, 2013.
- [5] C. Gao, D. Meng, Y. Yang, Y. Wang, X. Zhou, and A. G. Hauptmann, "Infrared patch-image model for small target detection in a single image," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4996–5009, 2013.
- [6] Y. Dai, Y. Wu, F. Zhou, and K. Barnard, "Attentional local contrast networks for infrared small target detection," *IEEE transactions on geoscience and remote sensing*, vol. 59, no. 11, pp. 9813–9824, 2021.
- [7] M. Zhang, R. Zhang, Y. Yang, H. Bai, J. Zhang, and J. Guo, "Isnet: Shape matters for infrared small target detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 877–886.
- [8] Y. Dai, Y. Wu, F. Zhou, and K. Barnard, "Asymmetric contextual modulation for infrared small target detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 950–959.
- [9] B. Li, C. Xiao, L. Wang, Y. Wang, Z. Lin, M. Li, W. An, and Y. Guo, "Dense nested attention network for infrared small target detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 1745–1758, 2022.
- [10] X. Wu, D. Hong, and J. Chanussot, "Uiu-net: U-net in u-net for infrared small object detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 364–376, 2022.
- [11] T. Zhang, L. Li, S. Cao, T. Pu, and Z. Peng, "Attention-guided pyramid context networks for detecting infrared small target under complex background," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 4, pp. 4250–4261, 2023.
- [12] Q. Liu, R. Liu, B. Zheng, H. Wang, and Y. Fu, "Infrared small target detection with scale and location sensitivity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 490–17 499.
- [13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.
- [14] L. Zhang, L. Peng, T. Zhang, S. Cao, and Z. Peng, "Infrared small target detection via non-convex rank approximation minimization joint l 2, l 1 norm," *Remote Sensing*, vol. 10, no. 11, p. 1821, 2018.
- [15] Y. Dai and Y. Wu, "Reweighted infrared patch-tensor model with both nonlocal and local priors for single-frame small target detection," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 10, no. 8, pp. 3752–3767, 2017.
- [16] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [17] F. Wu, T. Zhang, L. Li, Y. Huang, and Z. Peng, "Rpcanet: Deep unfolding rpca based infrared small target detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4809–4818.
- [18] F. Wu, Y. Dai, T. Zhang, Y. Ding, J. Yang, M.-M. Cheng, and Z. Peng, "Rpcanet++: Deep interpretable robust pca for sparse object segmentation," *arXiv preprint arXiv:2508.04190*, 2025.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.
- [20] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [21] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *International conference on machine learning*. PMLR, 2017, pp. 854–863.
- [22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.
- [23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 802–810.
- [24] L. Peng, T. Zhang, S. Huang, T. Pu, Y. Liu, Y. Lv, Y. Zheng, and Z. Peng, "Infrared small-target detection based on multi-directional multi-scale high-boost response," *Optical Review*, vol. 26, no. 6, pp. 568–582, 2019.
- [25] V. T. Tom, T. Peli, M. Leung, and J. E. Bondaryk, "Morphology-based algorithm for point target detection in infrared backgrounds," in *Signal and Data Processing of Small Targets 1993*, vol. 1954. International Society for Optics and Photonics, 1993, pp. 2–11.
- [26] S. D. Deshpande, M. H. Er, R. Venkateswarlu, and P. Chan, "Max-mean and max-median filters for detection of small targets," in *Signal and Data Processing of Small Targets 1999*, vol. 3809. International Society for Optics and Photonics, 1999, pp. 74–83.
- [27] Y. Wei, X. You, and H. Li, "Multiscale patch-based contrast measure for small infrared target detection," *Pattern Recognition*, vol. 58, pp. 216–226, 2016.
- [28] M. Zhang, Y. Wang, J. Guo, Y. Li, X. Gao, and J. Zhang, "Irsam: Advancing segment anything model for infrared small target detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 233–249.
- [29] F. Wu, A. Liu, T. Zhang, L. Zhang, J. Luo, and Z. Peng, "Saliency at the helm: Steering infrared small target detection with learnable kernels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–14, 2024.
- [30] J. Yang, S. Liu, J. Wu, X. Su, N. Hai, and X. Huang, "Pinwheel-shaped convolution and scale-based dynamic loss for infrared small

- target detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 9202–9210.
- [31] H. Li, J. Yang, R. Wang, and Y. Xu, “Ilnet: Low-level matters for salient infrared small target detection,” *IEEE Transactions on Aerospace and Electronic Systems*, 2025.
 - [32] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.
 - [33] T. Zhang, L. Li, and Z. Peng, “Optimization-inspired cumulative transmission network for image compressive sensing,” *Knowledge-Based Systems*, vol. 279, p. 110963, 2023.
 - [34] J. Zhang and B. Ghanem, “Ista-net: Interpretable optimization-inspired deep network for image compressive sensing,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 1828–1837.
 - [35] C. A. Metzler, A. Mousavi, and R. G. Baraniuk, “Learned D-AMP: Principled neural network based compressive image recovery,” 2017.
 - [36] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, “AMP-Net: Denoising based deep unfolding for compressive image sensing,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1487–1500, 2021.
 - [37] Y. Yang, J. Sun, H. Li, and Z. Xu, “Deep admm-net for compressive sensing mri,” in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.
 - [38] H. K. Aggarwal, M. P. Mani, and M. Jacob, “Modl: Model-based deep learning architecture for inverse problems,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2018.
 - [39] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a variational network for reconstruction of accelerated MRI data,” *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, 2017.
 - [40] J. Adler and O. Öktem, “Learned primal-dual reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.
 - [41] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson, “End-to-end variational networks for accelerated MRI reconstruction,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 64–73.
 - [42] H. Cai, J. Liu, and W. Yin, “Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 16977–16989.
 - [43] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” 2015.
 - [44] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” 2020.
 - [45] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.
 - [46] A. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” 2021.
 - [47] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” 2022.
 - [48] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations*, 2021.
 - [49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2022.
 - [50] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, ser. Applied Optimization. Springer, 2004, vol. 87.
 - [51] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
 - [52] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1724–1734.
 - [53] P. Putzky and M. Welling, “Recurrent inference machines for solving inverse problems,” 2017.
 - [54] S. A. H. Hosseini, B. Yaman, S. Moeller, M. Hong, and M. Akcakaya, “Dense recurrent neural networks for accelerated MRI: History-cognizant unrolling of optimization algorithms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1280–1291, 2020.
 - [55] G. Yiasemis, J.-J. Sonke, C. Sanchez, and J. Teuwen, “Recurrent variational network: A deep learning inverse problem solver applied to the task of accelerated MRI reconstruction,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 722–731.
 - [56] J. Song, B. Chen, and J. Zhang, “Memory-augmented deep unfolding network for compressive sensing,” in *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, 2021, pp. 4249–4258.
 - [57] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” 2022.
 - [58] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” 2024.
 - [59] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, “Vision mamba: Efficient visual representation learning with bidirectional state space model,” 2024.
 - [60] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, “Vmamba: Visual state space model,” 2024.
 - [61] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
 - [62] T. Zhang, H. Wu, Y. Liu, L. Peng, C. Yang, and Z. Peng, “Infrared small target detection based on non-convex optimization with lp-norm constraint,” *Remote Sensing*, vol. 11, no. 5, p. 559, 2019.
 - [63] S. Yuan, H. Qin, X. Yan, N. Akhtar, and A. Mian, “Sctransnet: Spatial-channel cross transformer network for infrared small target detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.
 - [64] Z. Xiong, F. Zhou, F. Wu, S. Yuan, M. Fu, Z. Peng, J. Yang, and Y. Dai, “Drpca-net: Make robust pca great again for infrared small target detection,” *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

APPENDIX A DERIVATION OF ADMM SUBPROBLEMS

This section derives the background, target, and noise subproblems from the augmented Lagrangian in Eq. 5. The dual and quadratic penalty terms in Eq. 5 can be merged by completing the square:

$$\begin{aligned} & \langle \mathcal{Y}, \mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N} \rangle + \frac{\mu}{2} \|\mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N}\|_F^2 \\ &= \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N} + \frac{1}{\mu} \mathcal{Y} \right\|_F^2 - \frac{1}{2\mu} \|\mathcal{Y}\|_F^2. \end{aligned} \quad (23)$$

Since the last term is independent of $\mathcal{B}$, $\mathcal{T}$, and $\mathcal{N}$ during primal-variable updates, it can be absorbed into a constant C . Therefore, the equivalent primal objective is

$$\begin{aligned} \mathcal{L}_\mu(\mathcal{B}, \mathcal{T}, \mathcal{N}, \mathcal{Y}) &= \mathcal{R}_\mathcal{B}(\mathcal{B}) + \mathcal{R}_\mathcal{T}(\mathcal{T}) + \mathcal{R}_\mathcal{N}(\mathcal{N}; \sigma) \\ &\quad + \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B} - \mathcal{T} - \mathcal{N} + \frac{1}{\mu} \mathcal{Y} \right\|_F^2 + C. \end{aligned} \quad (24)$$

ADMM then updates one latent component while fixing the others. For the background component:

$$\begin{aligned} \mathcal{B}^k &= \arg \min_{\mathcal{B}} \mathcal{R}_\mathcal{B}(\mathcal{B}) \\ &\quad + \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B} - \mathcal{T}^{k-1} - \mathcal{N}^{k-1} + \frac{1}{\mu} \mathcal{Y}^{k-1} \right\|_F^2. \end{aligned} \quad (25)$$

For the target component:

$$\begin{aligned} \mathcal{T}^k &= \arg \min_{\mathcal{T}} \mathcal{R}_\mathcal{T}(\mathcal{T}) \\ &\quad + \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B}^k - \mathcal{T} - \mathcal{N}^{k-1} + \frac{1}{\mu} \mathcal{Y}^{k-1} \right\|_F^2. \end{aligned} \quad (26)$$

For the noise component:

$$\begin{aligned} \mathcal{N}^k &= \arg \min_{\mathcal{N}} \mathcal{R}_\mathcal{N}(\mathcal{N}; \sigma) \\ &\quad + \frac{\mu}{2} \left\| \mathcal{X} - \mathcal{B}^k - \mathcal{T}^k - \mathcal{N} + \frac{1}{\mu} \mathcal{Y}^{k-1} \right\|_F^2. \end{aligned} \quad (27)$$

These three subproblems have the same latent structure: each branch minimizes an unknown latent regularizer plus a quadratic consistency term. This shared structure motivates the surrogate solver derived in the next section.

After the three primal variables are updated, ADMM updates the latent dual variable by accumulating the decomposition residual:

$$\mathcal{Y}^k = \mathcal{Y}^{k-1} + \mu (\mathcal{X} - \mathcal{B}^k - \mathcal{T}^k - \mathcal{N}^k). \quad (28)$$

APPENDIX B

DETAILED DERIVATION OF THE ADMM SURROGATE UPDATE

This section derives the analytical update of the $\mathcal{B}$ -subproblem under the Lipschitz-continuous gradient assumption. Starting from the background subproblem in Eq. 8, we first define

$$\mathcal{V}_B^{k-1} = \mathcal{X} - \mathcal{T}^{k-1} - \mathcal{N}^{k-1} + \frac{1}{\mu_B} \mathcal{Y}^{k-1}. \quad (29)$$

Then the $\mathcal{B}$ -subproblem can be written as

$$\mathcal{B}^k = \arg \min_{\mathcal{B}} \mathcal{R}_\mathcal{B}(\mathcal{B}) + \frac{\mu_B}{2} \|\mathcal{B} - \mathcal{V}_B^{k-1}\|_F^2. \quad (30)$$

Because hand-crafted constraints cannot be perfectly aligned with the physical prior of the latent background, we treat $\mathcal{R}_\mathcal{B}(\cdot)$ as an unknown regularizer rather than manually specifying its analytical form. We then use the Taylor upper-bound argument induced by the descent lemma to construct a quadratic majorization. Assume that $\nabla \mathcal{R}_\mathcal{B}$ is L_B -Lipschitz continuous, i.e.,

$$\|\nabla \mathcal{R}_\mathcal{B}(\mathcal{U}) - \nabla \mathcal{R}_\mathcal{B}(\mathcal{V})\|_F \leq L_B \|\mathcal{U} - \mathcal{V}\|_F. \quad (31)$$

Then $\mathcal{R}_\mathcal{B}(\mathcal{B})$ satisfies the following upper bound around $\mathcal{B}^{k-1}$:

$$\begin{aligned} \mathcal{R}_\mathcal{B}(\mathcal{B}) &\leq \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}) + \langle \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}), \mathcal{B} - \mathcal{B}^{k-1} \rangle \\ &\quad + \frac{L_B}{2} \|\mathcal{B} - \mathcal{B}^{k-1}\|_F^2, \quad \forall \mathcal{B} \in \mathbb{R}^{H \times W \times C}. \end{aligned} \quad (32)$$

Completing the square gives the equivalent surrogate regularizer

$$\hat{\mathcal{R}}_\mathcal{B}(\mathcal{B}, \mathcal{B}^{k-1}) = \frac{L_B}{2} \left\| \mathcal{B} - \left(\mathcal{B}^{k-1} - \frac{1}{L_B} \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}) \right) \right\|_F^2 + C_B, \quad (33)$$

where C_B is independent of $\mathcal{B}$ and therefore does not affect the minimizer. Substituting the surrogate regularizer in Eq. 33 into the $\mathcal{B}$ -subproblem in Eq. 30 gives

$$\begin{aligned} \hat{\mathcal{J}}_\mathcal{B}(\mathcal{B}) &= \frac{L_B}{2} \left\| \mathcal{B} - \left(\mathcal{B}^{k-1} - \frac{1}{L_B} \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}) \right) \right\|_F^2 \\ &\quad + \frac{\mu_B}{2} \|\mathcal{B} - \mathcal{V}_B^{k-1}\|_F^2. \end{aligned} \quad (34)$$

Since $\hat{\mathcal{J}}_\mathcal{B}(\mathcal{B})$ is the sum of two quadratic terms, we differentiate it with respect to $\mathcal{B}$:

$$\nabla \hat{\mathcal{J}}_\mathcal{B}(\mathcal{B}) = L_B \left(\mathcal{B} - \mathcal{B}^{k-1} + \frac{1}{L_B} \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}) \right) + \mu_B (\mathcal{B} - \mathcal{V}_B^{k-1}). \quad (35)$$

Setting $\nabla \hat{\mathcal{J}}_\mathcal{B}(\mathcal{B}) = 0$ yields

$$L_B \left(\mathcal{B} - \mathcal{B}^{k-1} + \frac{1}{L_B} \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}) \right) + \mu_B (\mathcal{B} - \mathcal{V}_B^{k-1}) = 0. \quad (36)$$

Rearranging the above equation gives the optimal solution

$$\mathcal{B}^k = \frac{L_B}{L_B + \mu_B} \mathcal{B}^{k-1} + \frac{\mu_B}{L_B + \mu_B} \mathcal{V}_B^{k-1} - \frac{1}{L_B + \mu_B} \nabla \mathcal{R}_\mathcal{B}(\mathcal{B}^{k-1}). \quad (37)$$

APPENDIX C

DERIVATION OF THE HIGH-ORDER MEMORY IN SOM

This section derives the high-order memory form of the Shared Optimization Memory (SOM). At stage k , SOM first constructs a joint optimization observation:

$$o_k = [\mathcal{B}^k, \mathcal{T}^k, \mathcal{N}^k, \mathcal{Y}^k, \mathcal{X}], \quad (38)$$

where o_k denotes the concatenated latent observation, $\mathcal{B}^k$, $\mathcal{T}^k$, and $\mathcal{N}^k$ are the background, target, and noise variables at stage k , $\mathcal{Y}^k$ is the latent dual variable, and $\mathcal{X}$ is the lifted infrared

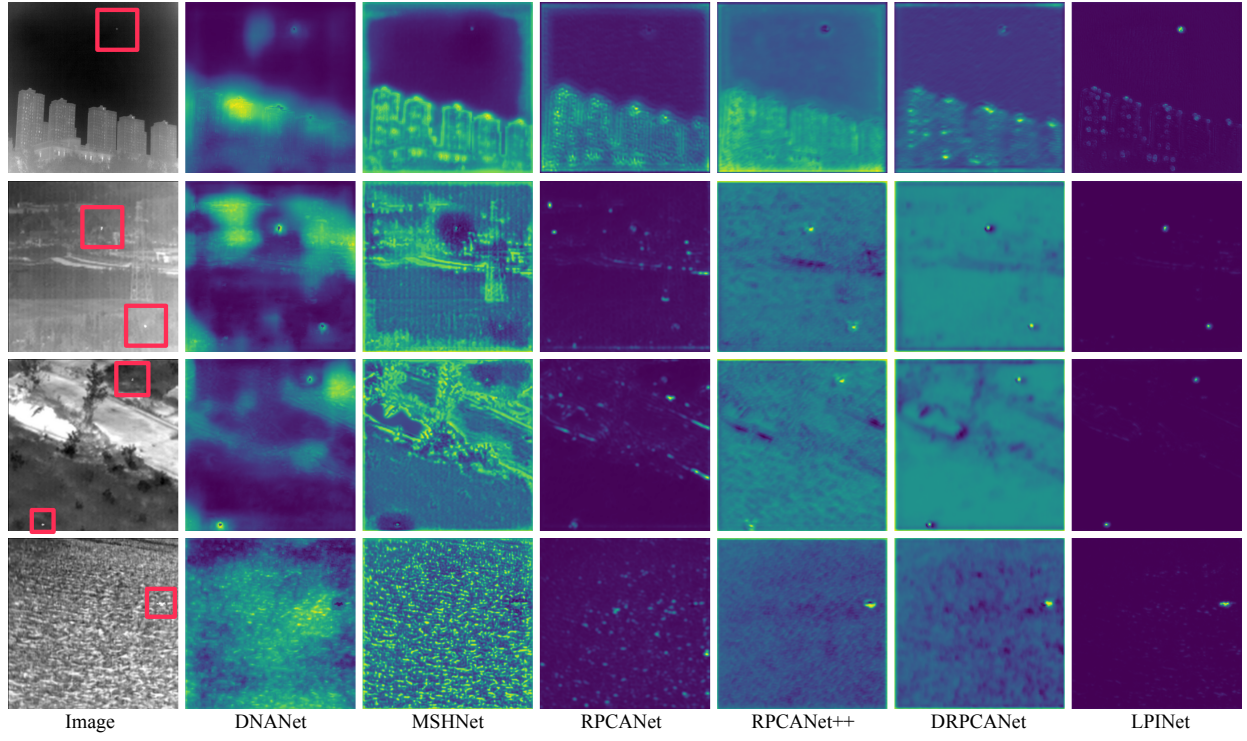


Fig. 12. Supplementary on the visualization of last layer feature maps produced by different methods. For each method, the feature map is projected by PCA, and the most informative principal component is selected for visualization. Rows 1 is *XDU67* in IRSTD-1k [7]. Row 2,3,4 are *000762 000410 000589* in NUDT-SIRST [9].

image representation. The optimization state cell updates the shared state h_k by

$$\begin{aligned} z_k &= \sigma(W_z * [o_k, h_{k-1}] + b_z), \\ r_k &= \sigma(W_r * [o_k, h_{k-1}] + b_r), \\ \tilde{h}_k &= \tanh(W_q * [o_k, r_k \odot h_{k-1}] + b_q), \\ h_k &= (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k. \end{aligned} \quad (39)$$

Here h_k is the shared optimization state at stage k . The functions $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and hyperbolic tangent activations. Since z_k , r_k , $\tilde{h}_k$, and h_k have compatible tensor shapes, all products below are element-wise.

The last line of Eq. 39 can be viewed as a gated interpolation between the previous state and the current candidate state. To expose its memory structure, we expand the state transition. For stage k , we have

$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k. \quad (40)$$

The previous state h_{k-1} follows the same recurrence:

$$h_{k-1} = (1 - z_{k-1}) \odot h_{k-2} + z_{k-1} \odot \tilde{h}_{k-1}. \quad (41)$$

Substituting Eq. 41 into Eq. 40 gives

$$\begin{aligned} h_k &= (1 - z_k) \odot [(1 - z_{k-1}) \odot h_{k-2} + z_{k-1} \odot \tilde{h}_{k-1}] \\ &\quad + z_k \odot \tilde{h}_k \\ &= (1 - z_k) \odot (1 - z_{k-1}) \odot h_{k-2} \\ &\quad + (1 - z_k) \odot z_{k-1} \odot \tilde{h}_{k-1} + z_k \odot \tilde{h}_k. \end{aligned} \quad (42)$$

Similarly, expanding h_{k-2} and substituting it into Eq. 42 yields

$$\begin{aligned} h_k &= (1 - z_k) \odot (1 - z_{k-1}) \odot (1 - z_{k-2}) \odot h_{k-3} \\ &\quad + (1 - z_k) \odot (1 - z_{k-1}) \odot z_{k-2} \odot \tilde{h}_{k-2} \\ &\quad + (1 - z_k) \odot z_{k-1} \odot \tilde{h}_{k-1} + z_k \odot \tilde{h}_k. \end{aligned} \quad (43)$$

Eq. 43 shows the pattern of the recursive expansion, each historical candidate $\tilde{h}_t$ is first injected through its own update gate z_t , and then propagated to stage k through the retention gates $(1 - z_{t+1}), \dots, (1 - z_k)$.

After expanding the recurrence for m steps, where $1 \leq m \leq k$, the state can be written as:

$$\begin{aligned} h_k &= \left(\prod_{i=k-m+1}^k (1 - z_i) \right) \odot h_{k-m} \\ &\quad + \sum_{t=k-m+1}^k \left[z_t \odot \tilde{h}_t \odot \prod_{i=t+1}^k (1 - z_i) \right]. \end{aligned} \quad (44)$$

When $t = k$, the product $\prod_{i=k+1}^k (1 - z_i)$ is an empty product and is defined as an all-one tensor with the same shape as h_k , so the last term becomes $z_k \odot \tilde{h}_k$. Setting $m = k$ expands the recurrence back to the initial state h_0 :

$$h_k = \left(\prod_{i=1}^k (1 - z_i) \right) \odot h_0 + \sum_{t=1}^k \left[z_t \odot \tilde{h}_t \odot \prod_{i=t+1}^k (1 - z_i) \right]. \quad (45)$$

This is the same expansion as Eq. 20. In this expression, the contribution of each candidate state $\tilde{h}_t$ is controlled by its update gate z_t and the subsequent retention gates $\prod_{i=t+1}^k (1 - z_i)$.

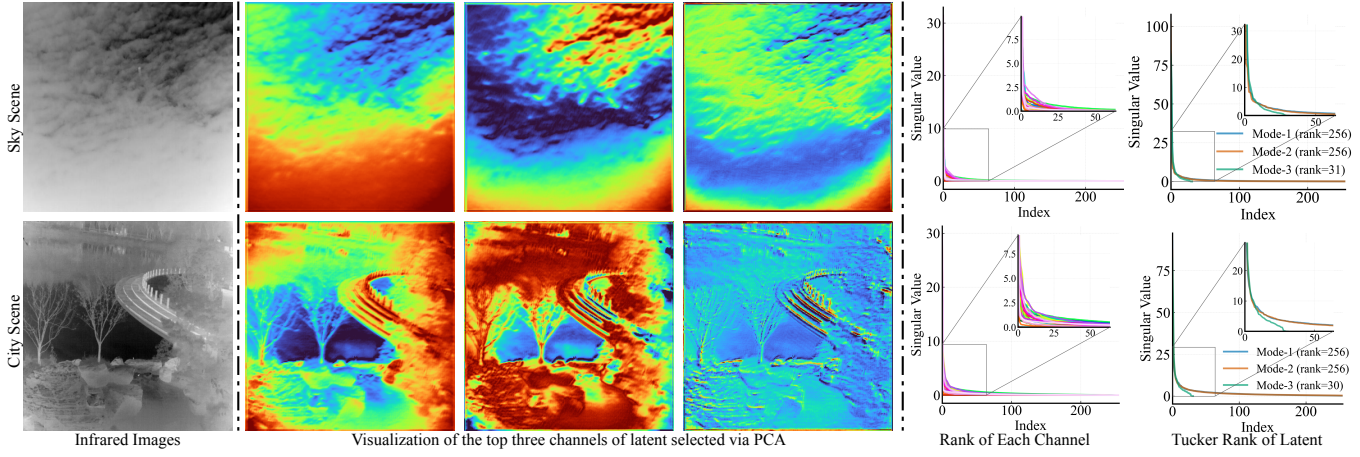


Fig. 13. Visualization of the low-rank property in latent domain. From left to right are the original infrared image, a visualized top three channels of the latent selected via PCA, rank of each channel and Tucker rank analysis of the latent. It can be observed that the latent exhibits distinct low-rank property in both visual visualization and rank analysis. The images include sky scene (Top *XDU145* in IRSTD-1k [7]) and ground scene (bottom *XDU202*).

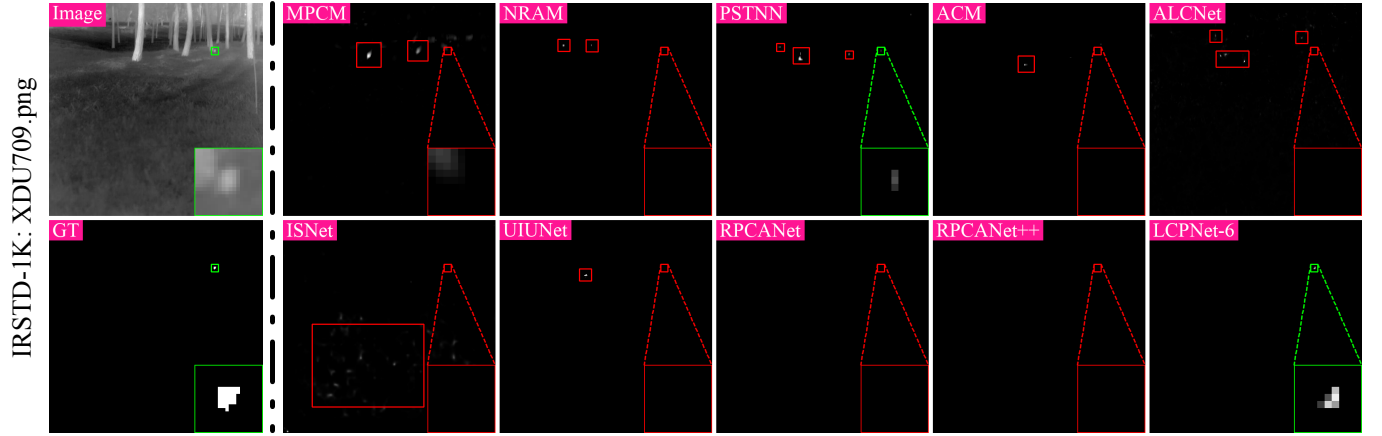


Fig. 14. Qualitative comparison on image *XDU709*. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

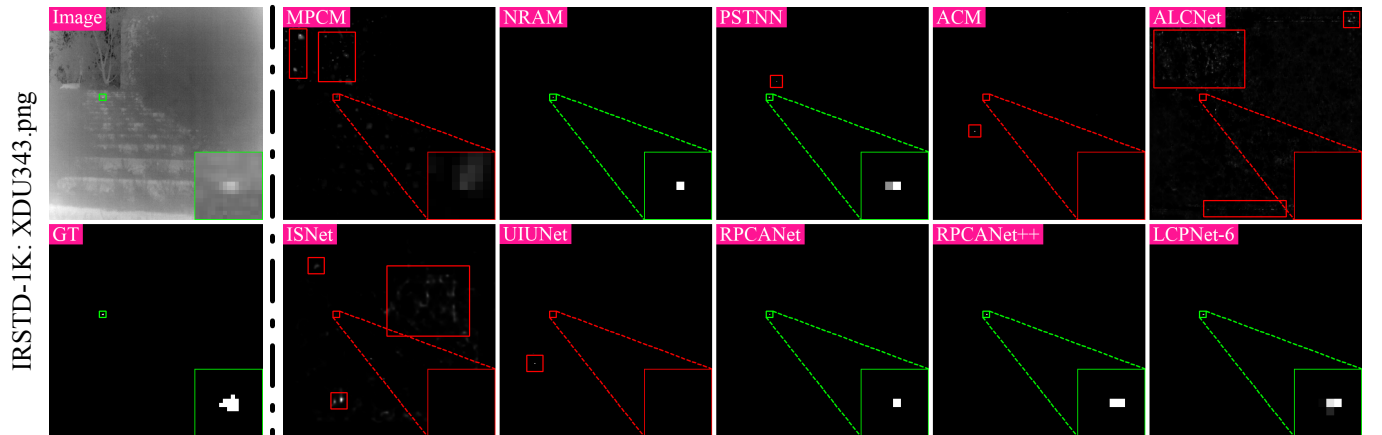


Fig. 15. Qualitative comparison on image *XDU343*. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

Therefore, h_k is not determined only by h_{k-1} or a fixed short window of previous states. Instead, it contains a gated sum of all candidate states $\{\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_k\}$ together with the retained initial state h_0 . Thus, SOM has an unbounded effective memory order in its recurrent formulation, while

the actual contribution of each historical state is adaptively controlled by the update gates.

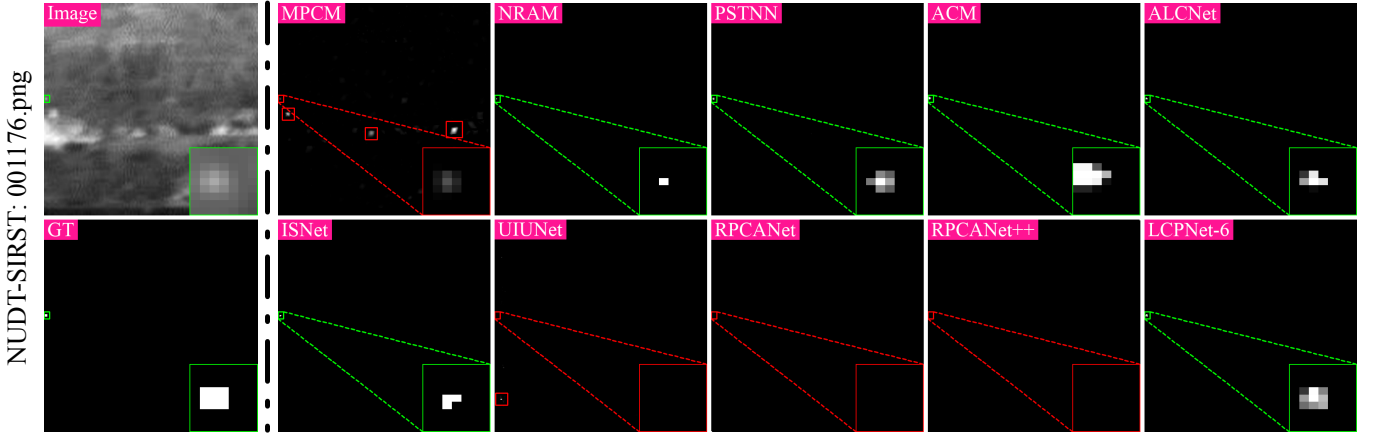


Fig. 16. Qualitative comparison on image 001176. Green boxes denote correct detections, while red boxes denote false alarms or missed targets. Best view in color.

APPENDIX D MORE EXPERIMENTAL RESULTS

This section provides additional qualitative results to complement the visual comparisons in the main paper. Fig. 13 further visualizes the low-rank property of the latent representation, where the latent features exhibit distinct low-rank characteristics in both visual and rank analysis. Fig. 12 further visualizes the last-layer feature responses, where LCPNet shows more concentrated target-related activations and weaker background interference in challenging scenes. Figs. 14–16 present additional detection examples with different target appearances and background structures. Across these cases, LCPNet maintains compact responses around true targets and produces fewer clutter-induced false alarms, further supporting the robustness of the proposed latent-state unfolding design.