LIWI: LAYERING IN THE WILD

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in generative models have empowered impressive layered image generation, yet their success is largely confined to graphic design domains. The layering of in-the-wild images remains an underexplored problem, limiting fine-grained editing and applications of images in real-world scenarios. Specifically, challenges remain in scalable layered data and the modeling of object interaction in natural images, such as illumination effects and structural boundary. To address these bottlenecks, we propose a novel framework for high-fidelity natural image decomposition. First, we introduce an *Agent-driven Data Decomposition* (ADD) pipeline that orchestrates agents and tools to synthesize layered data without manual intervention. Utilizing this pipeline, we construct a large-scale dataset, named LiWi-100k, with over 100,000 high-quality layered in-the-wild images. Second, we present a novel framework that jointly improves photometric fidelity and alpha boundary accuracy. Specifically, photometric residual prediction models differences unexplained by RGBA compositing, and degradation-restoration objective provides boundary-correction supervision by recovering clean foreground image from degraded one. Extensive experiments demonstrate that our framework achieves state-of-the-art (SoTA) performance in natural image decomposition, outperforming existing models in RGB L1 and Alpha IoU metrics. We will soon release our code and dataset.

1 INTRODUCTION

Layer decomposition aims to convert a flattened image into a set of visual elements, such as foreground objects with their alpha masks and a clean background. It unlocks essential structural priors required for controllable video generation (*e.g.*, independent foreground-background motion), 3D asset synthesis (*e.g.*, cues for occluded entities), and the development of interactive world models (Kasten et al., 2021; Bar-Tal et al., 2022; Lee et al., 2023). Compared with conventional segmentation or matting, image layering requires not only identifying visible object regions but also recovering complete layer appearances and the scene content behind them (Suvorov et al., 2022). This makes in-the-wild image layering a useful intermediate representation between pixel-level image generation and structured visual understanding.

Despite recent progress in layered image generation and decomposition, most existing methods (Suzuki et al., 2025; Yin et al., 2025; Liu et al., 2025) are mostly compatible to the decomposition of graphic design, PSD (Photoshop Document) assets, or synthetically composed images. These domains usually contain clean boundaries, explicit layer ordering, and simple alpha blending. Real photographs are more challenging. Foreground objects do not merely occlude the background; they also change the scene through cast shadows, contact darkening, reflections, soft boundaries, and local illumination variations (Garces et al., 2022). The target of natural image layering is not only to separate visible elements but also to decide where the physical traces caused by those objects should go. As a result, a real-world image cannot be fully explained by simply stacking RGBA layers.

A central obstacle of natural image decomposition is the lack of training data for in-the-wild image layering task. Unlike graphic designs, real-world images do not provide authored layers. Manually annotating is expensive and difficult to scale. To address this data bottleneck, we propose the ADD pipeline that constructs layered supervision from in-the-wild images without manual annotation. ADD enables agents and specialized tools to generate clean backgrounds, complete foreground RGBA layers, and select consistent layer combinations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

However, high-quality layered data alone is not sufficient for natural photographs. Real-world scenes are governed by complex illumination. Effects such as shadows and lighting variations are contextual footprints induced by foreground objects and background scenes. We therefore introduce a photometric residual layer to explicitly represent the difference between the source image and the recomposed image, including illumination effects and shadows. Instead of forcing such residuals to be ambiguously absorbed by the foreground or background, the photometric residual layer provides supervision for global illumination interactions. This encourages the model to disentangle visual traces induced by foreground entities.

Beyond color fidelity, in-the-wild image layering also requires accurate layer boundaries. We observe that many failure cases arise from local boundary degradation, including mask erosion, slight dilation, and inaccurate color blending near object contours. To address these boundary-level errors, we introduce a degradation-restoration objective as an auxiliary foreground refinement task. During training, foreground layers are deliberately corrupted, and the model is trained to recover the corresponding clean layers. This restoration-oriented supervision encourages the model to capture the mechanisms behind alpha boundary formation, local color correction, and texture preservation.

Our main contributions are summarized as follows: (1) We propose the ADD pipeline and construct LiWi-100k, a large-scale and high-quality dataset for in-the-wild image layering, eliminating the need for expensive manual annotation. (2) We propose a layer decomposition framework that combines photometric residual prediction with auxiliary layer refinement. The residual captures photometric variations, while the degradation-restoration objective improves boundary accuracy. (3) Extensive experiments demonstrate that our framework achieves SoTA performance both on LiWi-100k and Crello (Yamaguchi, 2021), outperforming existing models in RGB L1 and Alpha IoU.

2 Related Work

2.1 Image Layer Decomposition

Layered image decomposition provides an interpretable representation for image editing, compositional generation, and inverse graphics (Huang et al., 2018; Yang et al., 2025). Recent work has advanced from synthetic compositions to editable full-RGBA representations (Zhang et al., 2023; Huang et al., 2024; Pu et al., 2025), including matting-based data construction in Text2Layer (Zhang et al., 2023), modular open-domain decomposition in MULAN (Tudosiu et al., 2024), iterative top-layer extraction for graphic designs in LayerD (Suzuki et al., 2025), and end-to-end diffusion-based RGB-to-RGBA decomposition in Qwen-Image-Layered (Yin et al., 2025; Wu et al., 2025). However, natural-image decomposition remains difficult: object-centric pipelines accumulate errors across intermediate modules (Tudosiu et al., 2024), design-oriented methods assume clean boundaries and organized layers rarely found in photographs (Suzuki et al., 2025; Chen et al., 2025b), and recent end-to-end approaches are trained on PSD-like authoring data, making them better suited to design-style semantic layers than to natural-scene photometry (Yin et al., 2025; Liu et al., 2025). Real photographs involve entangled shadows, reflections, translucency, soft transitions, and occlusions, which complicate both layer separation and cross-layer interaction modeling (Yang et al., 2026; Chen et al., 2026; Yang et al., 2025). We address this gap by decomposing natural images with a training strategy that better preserves photometric effects and compositional interactions.

2.2 RGBA Dataset Construction

Training data for layered image modeling typically follows two routes: synthetic composition, which composites foregrounds, masks, or transparent layers under predefined blending rules to provide scalable multilayer supervision with explicit control over layer order and alpha blending (Zhang & Agrawala, 2024; Huang et al., 2025; 2026); and extraction-based pipelines, which derive foregrounds from segmentation or matting, reconstruct backgrounds via inpainting, and infer layer order from geometric or learned cues (Kang et al., 2025; Yang et al., 2025). However, both remain insufficient for natural-image layer decomposition. Synthetic data often exhibits overly clean interactions and a realism gap (Dalva et al., 2024; Chen et al., 2025a), whereas extraction-based pipelines are vulnerable to upstream errors and accumulate structural and photometric artifacts across stages (Kang et al., 2025). Agent-style automation can improve scalability, but tightly coupled multi-stage workflows remain brittle when multiple dependent decisions must be jointly correct (Chen et al., 2026; Sun et al.,

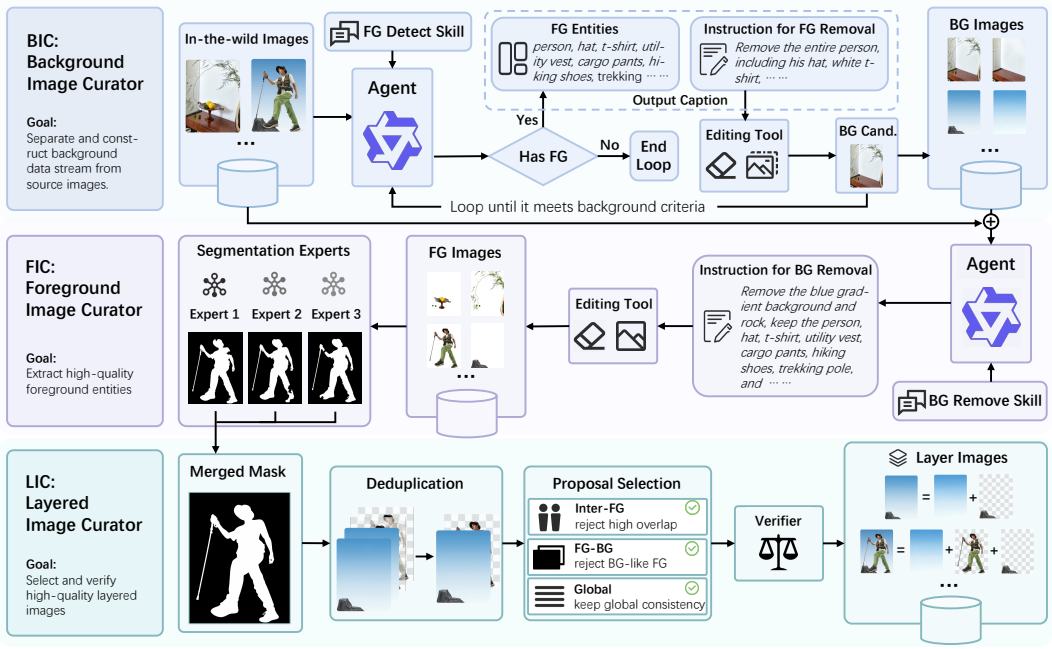

Figure 1: Overview of our ADD pipeline. The system leverages agent and specialized tools to automatically decompose in-the-wild images. Foreground and background layers are routed into separate repositories and subsequently selected by the LIC module, where a rigorous verifier ensures the quality of the final layered compositions.

2025). To address these limitations, we propose a decoupled data construction pipeline that separately builds backgrounds, foregrounds, and final layered composites, reducing inter-stage interference while preserving layered consistency and photometric realism. A consensus-based verification mechanism further filters unreliable samples, enabling a more scalable and reliable dataset for natural-image layer decomposition.

3 Synthesizing Layered Images in the Wild

Learning in-the-wild image layering requires supervision that is rarely available in real photographs. Unlike graphic designs or PSD files, where layers are explicitly authored, an in-the-wild image only provides a flattened RGB observation in which foreground appearance, occlusion, cast shadows, reflections, and illumination changes are entangled. A simple segmentation mask can recover the visible foreground region, but it does not reveal the clean background behind the object, nor does it explain the photometric footprint left by the foreground on the scene. To address these problems in layering task, we introduce the ADD pipeline, a multi-agent system that automatically synthesizes high-quality layered samples from in-the-wild images.

3.1 Problem Formulation and Overview

Given a collection of in-the-wild images $\mathcal{I}$, our goal is to construct a layered dataset $\mathcal{D} = \{(I_{src}, B, \{F_k, \alpha_k\}_{k=1}^{K})\}$, where I_{src} is the input image to be decomposed into background image B and foreground images $\{F_k, \alpha_k\}_{k=1}^{K}$. F_k and α_k denote the RGB appearance and alpha mask of the k-th foreground image. Note that I_{src} can be original image from $\mathcal{I}$ or intermediate background in data curation. The key requirement of layered images is that all components should be both individually valid and jointly consistent. Specifically, foreground entities should be complete and semantically meaningful, background should be free of foreground artifacts, and their composition I_{src} should preserve plausible spatial and photometric interactions.

As shown in Fig. 1, the proposed ADD is implemented as an agentic system and contains three collaborative curators: the *Background Image Curator* (BIC), the *Foreground Image Curator* (FIC),

Algorithm 1: Proposal Selector for LIC

Require: input image I; backgrounds $\mathcal{B} = \{B_k\}_{k=1}^{K}$; foregrounds $\mathcal{F} = \{(\tilde{F}_k, \alpha_k)\}_{k=1}^{K}$; DINOv2 $\phi(\cdot)$; thresholds $\tau_{local}, \tau_{global}$.
Ensure: proposals of layered images $\mathcal{P}$.

$M(\alpha, X) \triangleq X \odot \alpha + \mathbf{1}(1-\alpha)$ ▷ Function Definition
$f_{ij}^{F} \leftarrow \phi(M(\alpha_i, \tilde{F}_j)), \forall i, j \in [1, 2, ..., K]$
$f_{ij}^{B} \leftarrow \phi(M(\alpha_i, B_j)), \forall i, j \in [1, 2, ..., K]$
$\mathcal{P} \leftarrow \varnothing, \mathcal{F}_{valid} \leftarrow \varnothing$
for $\mathcal{F}_{sub} \subseteq \mathcal{F}$ **do** ▷ Inter-FG Overlap
  if $\exists F_i, F_j \in \mathcal{F}_{sub}, i < j$ and $\langle f_{ii}^{F}, f_{ij}^{F} \rangle > \tau_{local}$
    continue
  else
    $\mathcal{F}_{valid} \leftarrow \{\mathcal{F}_{sub}\} \cup \mathcal{F}_{valid}$
end for
for $I_{src} \in \{I\} \cup \mathcal{B}$, $B_j \in \mathcal{B} \setminus \{I_{src}\}$, $\mathcal{F}_{sub} \subseteq \mathcal{F}_{valid}$ **do**
  if $\exists F_i \in \mathcal{F}_{sub}, \langle f_{ii}^{F}, f_{ij}^{B} \rangle > \tau_{local}$ ▷ FG-BG Overlap
    continue
  $I_c \leftarrow \textsc{Composite}(B_j, \mathcal{F}_{sub})$
  if $\langle \phi(I_c), \phi(I_{src}) \rangle \geq \tau_{global}$ ▷ Global Consistency
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(I_{src}, B_j, \mathcal{F}_{sub})\}$
end for
return $\mathcal{P}$

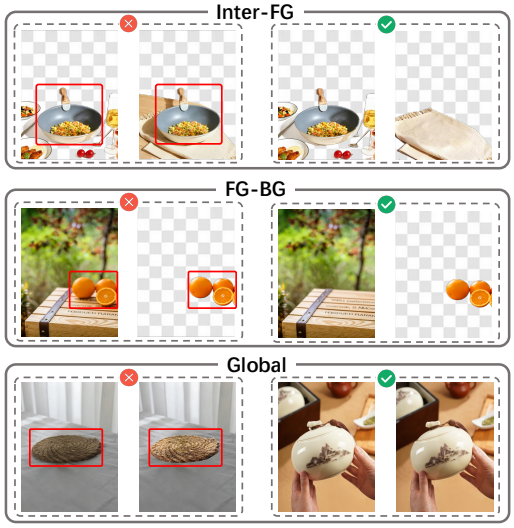

Figure 2: Illustration of pass and fail examples from Inter-FG, FG-BG and Global Consistency constraints.

and the *Layered Image Curator* (LIC). BIC builds a repository of clean backgrounds, FIC extracts high-quality foreground entities with transparent masks, and LIC selects compatible foreground-background combinations to produce final layered samples. This agent-driven mechanism enables scalable data construction while avoiding the requirement for manual intervention.

3.2 Background and Foreground Curation

Given an image $I \in \mathcal{I}$, the BIC constructs a pool of background candidates $\mathcal{B}$ in a loop starting from $I_0 = I$. In the i-th ($i \geq 0$) step, the agent first detects whether there is foreground entity in the input I_i by foreground detection skill. If no foreground is detected, the loop ends. Otherwise, the agent generates an editing instruction that describes the complete foreground region to be removed, including the main object, accessories, and visually attached parts. The agent then calls an editing tool to produce a background candidate B_{i+1}, which is set to I_{i+1} for the next step, based on the foreground removal instruction. Note that the foreground descriptions are reusable in FIC.

Based on raw image I and background candidates $\mathcal{B}$, the FIC builds a foreground repository $\mathcal{F}$ containing complete foreground entities. Rethinking the i-th step of BIC, the dominated foreground entities can be detected in the input image I_i where $i \in \{0, 1, ..., |\mathcal{B}| - 1\}$. With the detected foreground entities, the agent generates a background removal instruction by specifying the retained foreground entities and all visually attached components. The editing tool then erases the surrounding background and produces a foreground image with white background, denoted as $\tilde{F}_{i+1}$, for simple segmentation. Since a single segmentation may produce incomplete mask around the regions of thin structures, accessories, or boundaries, we use N segmentation experts to obtain candidate masks $\{M_{i+1}^{(1)}, M_{i+1}^{(2)}, \ldots, M_{i+1}^{(N)}\}$ from $\tilde{F}_{i+1}$. We merge these candidate masks to generate $\alpha_{i+1} = \text{avg}(M_{i+1}^{(1)}, M_{i+1}^{(2)}, \ldots, M_{i+1}^{(N)})$. The final RGBA foreground image F_{i+1} is constructed by merging the RGB content of $\tilde{F}_{i+1}$ with the alpha map α_{i+1}. In this way, the foreground extraction is simplified by first removing the complex background context and then using multi-expert mask fusion to obtain a complete alpha map.

3.3 Layered Composition and Verification

Given the background pool $\mathcal{B}$ produced by BIC and the RGBA foreground pool $\mathcal{F}$ produced by FIC, LIC de-duplicates the near-identical images within $\mathcal{B}$ and $\mathcal{F}$ respectively. This avoids excessive visual redundancy in the selection process of foreground-background compositions.

After de-duplication, LIC selects the compatible foreground-background compositions by verifying the full combinations of candidates. As summarized in Alg. 1 and illustration of passed and failure

examples in Fig. 2, the Proposal Selector evaluates every combination from three perspectives. First, the *Inter-FG* constraint removes foreground combinations with strong overlap or semantic redundancy. For each foreground pair (F_i, F_j) with $i < j$ (*i.e.*, F_i is in front of F_j and can occlude it), we use the mask of F_i (*i.e.*, α_i) to cut out the corresponding region of F_j. If the similarity is abnormally high, the two layers are likely to describe the same object or heavily occluded regions, and the candidate is rejected. Second, the *FG-BG* constraint checks whether a background contains the foreground contents. If the masked background region is highly similar to the foreground itself, the background is likely to retain redundant elements and should be discarded. For proposals that pass the entity-level checks, the *Global Consistency* constraint evaluates the rendered composition I_c. The holistic consistency of I_c is then measured against source images I_{src}. Only compositions that maintain a high global similarity score are retained, ensuring that the selected layers form a plausible natural image rather than an arbitrary collage. In practice, we use DINOv2 (Oquab et al., 2024) embeddings to represent images; Appendix H compares it with pixel-wise distance.

The remaining composition candidates are further examined by a model verifier. The verifier checks both layer-wise quality and composition-level validity, including foreground completeness, background cleanliness, absence of obvious artifacts, and the semantic plausibility of the rendered image. Candidates that fail these checks are rejected, while accepted candidates are stored as final layered samples. Through this selector-verifier design, LIC automatically transforms independent background and foreground streams into consistent layered images, providing scalable, human-free supervision for in-the-wild images.

3.4 LiWi-100k Dataset

We introduce LiWi-100k, a layered dataset dedicated exclusively to real-world scenes. It contains 100,371 high-quality layered images built entirely from unstructured real-world images without manual annotation. Although these layers serve as pseudo-ground truth, they exhibit high reliability, achieving a 94.9% human acceptance rate in our manual quality assessment (see Appendix A for details). The curation process is fully automated through our proposed ADD pipeline, which orchestrates a suite of open-source models. The Agent and Verifier are instantiated with Qwen3-VL-32B (Bai et al., 2025), generating precise removal instructions and verifying the proposals. The Editing Tool is powered by FLUX.2-klein-9B (Black Forest Labs, 2026), ensuring high-fidelity removal for background and foreground. For segmentation, we employ an ensemble of experts comprising RMBG-1.4 (BRIA AI, 2024a), RMBG-2.0 (Zheng et al., 2024a; BRIA AI, 2024b), and SAM3 (Carion et al., 2025). A comparison with a MuLAn-style (Tudosiu et al., 2024) pipeline and an analysis of VLM sensitivity are provided in Appendices F and G, respectively.

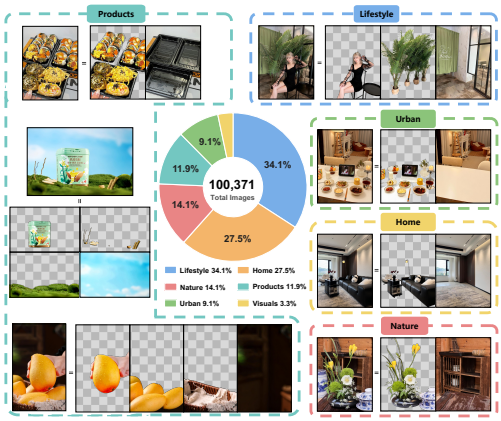

Figure 3: Data distribution and samples of LiWi-100k.

As illustrated in Fig. 3, LiWi-100k covers diverse real-world categories. The complete distribution across six broad categories and 16 fine-grained categories is provided in Appendix D. Regarding structural complexity, the dataset contains a maximum of 5 layers. The vast majority of the samples (55%) consist of 2 layers, while the remaining 45% contain 3 to 5 layers.

4 LiWi Framework

This section presents our end-to-end layer-decomposition model **LiWi** trained on LiWi-100k. LiWi replaces source-image reconstruction with photometric residual prediction and adds a degradation-restoration path used only during training to refine foreground boundaries.

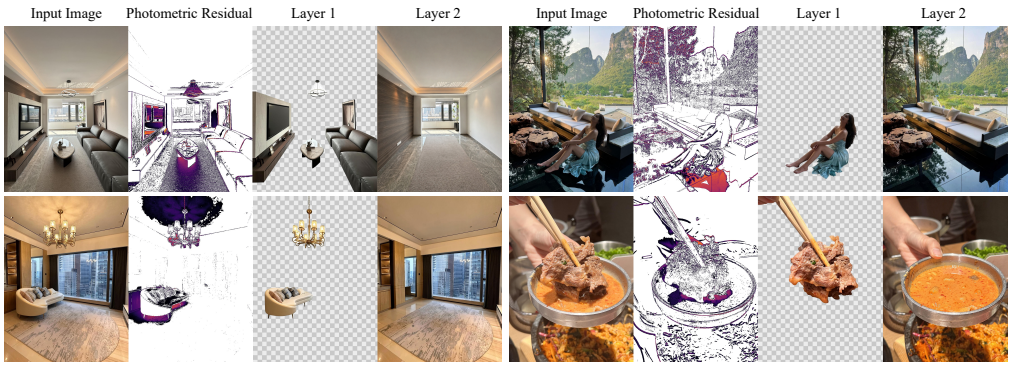

Figure 4: Visualization of the photometric residual layer. It records foreground-related lighting changes, such as shadows and occlusion, helping the model remove them when restoring a clean background.

4.1 Photometric Residual Learning

Real-world photographs contain complex photometric effects, such as cast shadows, illumination variations, and contact darkening. As shown in Fig. 4, we introduce a photometric residual layer to capture differences unexplained by RGBA compositing. Specifically, let I_c denote the recomposed image, obtained by stacking the background B and all foreground layers $\{F_k\}_{k=1}^{K}$ in order. The photometric residual layer is defined as $R = I_{src} - I_c$. It can contain lighting effects, reflections, alpha bleeding, and shadows of the foreground objects. Instead of forcing the illumination changes to be ambiguously absorbed by either the foreground or the background layer, we explicitly model the photometric residual layer.

During the training process of diffusion model, rather than regenerating the source image I_{src} (Yin et al., 2025), we propose to model the generation process of photometric residual layer R, which avoids the arbitrary information propagation of artifacts. To validate the effectiveness of this design, we analyze the attention weights of the noised I_{src} or R to other input tokens (*i.e.*, clean I_{src} and noised layers). When reconstructing I_{src} (Fig. 5, top), the model predominantly attends to the original input image. This may indicate the information leakage from clean reference image I_{src} to noised I_{src}. When the objective is shifted to predicting R (Fig. 5, bottom), the attention distribution becomes more balanced among the clean I_{src} and layered images.

By using the photometric residual layer to absorb the complex illumination variations, we prevent lighting artifacts from being erroneously attached to the layered images, and thus encourage layered images to concentrate on the generation of foreground entities and background scenes. Therefore, the network successfully decouples the foreground entities and achieves improved accuracy in color consistency.



Figure 5: Attention-weight comparison between different reconstruction objectives.

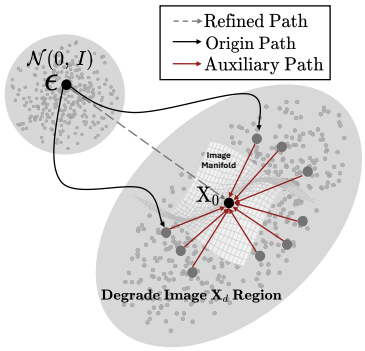

Figure 6: Illustration of the restoration process from degraded regions to the natural image manifold.

4.2 Degraded Boundary Refinement

In the layer generation task, given the ground-truth image $x_0 \in \{R\} \cup \mathcal{B} \cup \mathcal{F}$, the flow-matching (Lipman et al., 2022) method constructs a linear path that transports a Gaussian sample ϵ to image x_0. The latent representation at time step $t \in [0, 1]$ is defined via linear interpolation:

$$z_t = (1 - t)\epsilon + tx_0. \tag{1}$$

However, natural images often contain complex structures of objects, leading to degraded boundary artifacts in the foreground generation. To refine the artifacts of generated foreground, we explicitly model the boundary refinement task in the diffusion model. Specifically, we construct the degraded image x_d from the ground-truth image $x_0 \in \mathcal{F}$ by erosion, dilation, or blurring of the boundary. an auxiliary flow path is introduced to transport the noised degradation image $x_d + \epsilon$ to the ground-truth image x_0, which yields the auxiliary path as:

$$z_t^{aux} = (1 - t)(x_d + \epsilon) + tx_0. \tag{2}$$

As illustrated in Fig. 6, we shift the start-point of Gaussian noise ϵ to the degraded observation $x_d + \epsilon$ around the ground-truth image x_0, expanding the exploration path for degraded boundary refinement. This auxiliary path shares the same model weights with the original flow path in the foreground generation, and thus provides an additional supervision for boundary correction. The final training objective combines the original flow matching loss and the auxiliary boundary-correction loss:

$$\mathcal{L} = \mathbb{E}_{t,x_0\in\{R\}\cup\mathcal{B}\cup\mathcal{F},\epsilon}\left[|v_\theta(z_t, t) - v_t|_2^2\right] + \lambda\mathbb{E}_{t,x_0\in\mathcal{F},x_d,\epsilon}\left[|v_\theta(z_t^{aux}, t) - v_t^{aux}|_2^2\right], \tag{3}$$

where $v_t = x_0 - \epsilon$, $v_t^{aux} = x_0 - x_d - \epsilon$, and $\lambda = 0.1$ controls the strength of auxiliary supervision. In implementation, we update the attention mask and position embeddings due to the additional input of x_d. For each degraded image, we use the same position embedding as its corresponding foreground image. In attention layer, each degraded image only attends to itself and the source image I_{src}. Therefore, the model learns both noise-to-image generation and degraded boundary correction, leading to more accurate boundaries. During inference, we use the original flow path for layer generation, while the auxiliary path is only used as an additional training objective.

5 Experiments

5.1 Experimental Setup

Implementation Details. We initialize from Qwen-Image-Layered (Yin et al., 2025) and train on LiWi-100k for 12K steps using Adam (Kingma & Ba, 2014), a constant learning rate of 1×10^{-5}, 16 NVIDIA B200 GPUs, and batch size 16. Samples are bucketed by aspect ratio and layer count, with resolution capped at 1024×1024. LiWi-Crello is further trained on Crello for 6K steps with the same settings. Computational costs are reported in Appendix E.

Datasets and Metrics. We evaluate our framework on two benchmarks: the proposed LiWi-100k and the Crello (Yamaguchi, 2021) test set. LiWi-100k contains 1,000 in-the-wild test images. Training and test sets are split by the original raw images used for data construction, with all samples derived from the same original image assigned to the same split. The Crello test set consists of 1,972 raster graphic design templates. Following LayerD and Qwen-Image-Layered (Suzuki et al., 2025; Yin et al., 2025), we report RGB L1 and Alpha soft IoU. RGB L1 measures RGB reconstruction accuracy, with lower values indicating better color and texture fidelity. Alpha soft IoU computes IoU on continuous alpha values, with higher values indicating more accurate opacity and boundary estimation. Max Edits denotes the maximum number of adjacent-layer merges allowed between predicted and reference layer sequences, with zero allowing no merges. See Appendix B for details.

Baselines. On LiWi-100k, we compare with LayerD (Suzuki et al., 2025), Qwen-Image-Layered, its variant fine-tuned on LiWi-100k, and two iterative modular pipelines. Both modular pipelines use Qwen3-VL-32B (Bai et al., 2025) to identify foreground objects, SAM3 (Carion et al., 2025) to estimate masks, and Matting-Anything (Li et al., 2024) to refine them, followed by either FLUX.2-klein-9B (Black Forest Labs, 2026) or Qwen-Image-Edit (Wu et al., 2025) for inpainting. The process repeats from front to back until the target number of layers is reached.

Table 1: Quantitative results on the original LiWi-100k test set.

Metric	RGB L1 ↓						Alpha soft IoU ↑					
# Max Edits	0	1	2	3	4	5	0	1	2	3	4	5
LayerD (Suzuki et al., 2025)	0.1342	0.1341	0.1338	0.1336	0.1335	0.1333	0.5926	0.6007	0.6010	0.6015	0.6019	0.6022
Modular baseline (FLUX.2-klein-9B)	0.1787	0.1781	0.1772	0.1766	0.1760	0.1756	0.7277	0.7359	0.7433	0.7494	0.7544	0.7582
Modular baseline (Qwen-Image-Edit)	0.1649	0.1641	0.1608	0.1588	0.1571	0.1558	0.7242	0.7251	0.7357	0.7419	0.7469	0.7507
Qwen-Image-Layered (Yin et al., 2025)	0.2692	0.2666	0.2650	0.2635	0.2622	0.2613	0.5878	0.5932	0.5979	0.6018	0.6050	0.6075
Qwen-Image-Layered-SFT	0.0996	0.0993	0.0974	0.0963	0.0953	0.0946	0.8745	0.8754	0.8783	0.8801	0.8816	0.8828
LiWi	**0.0907**	**0.0906**	**0.0895**	**0.0889**	**0.0883**	**0.0879**	**0.9314**	**0.9319**	**0.9332**	**0.9340**	**0.9347**	**0.9353**

5.2 Quantitative Results

Layer Decomposition. We report quantitative comparisons in Tables 1 and 2. On LiWi-100k, the original Qwen-Image-Layered model shows a clear domain gap when transferred from graphic designs to in-the-wild images. Qwen-Image-Layered-SFT which is fine-tuned on our data improves both RGB reconstruction and alpha estimation. Compared with Qwen-Image-Layered-SFT, LiWi reduces RGB L1 by 7.99% on average and improves alpha IoU by 6.22%. Particularly, the modular pipelines exhibit larger RGB errors in inpainted regions. On the Crello (Yamaguchi, 2021) benchmark, LiWi also outperforms prior methods. Although Crello contains raster graphic designs rather than natural photographs, LiWi reduces RGB L1 by 12.45% on average over Qwen-Image-Layered and improves alpha soft IoU by 1.35%. These results show that LiWi achieves strong gains on in-the-wild images while retaining robust performance on the dataset that do not rely on ADD labels

Table 2: Evaluation on Crello (Yamaguchi, 2021) test set under different maximum edit numbers.

Metric	RGB L1 ↓						Alpha soft IoU ↑					
# Max Edits	0	1	2	3	4	5	0	1	2	3	4	5
LayerD	0.0709	0.0541	0.0457	0.0419	0.0403	0.0396	0.7520	0.8111	0.8435	0.8564	0.8622	0.8650
Qwen-Image-Layered	0.0594	0.0490	0.0393	0.0377	0.0364	0.0363	0.8705	0.8863	0.9105	0.9121	0.9156	0.9160
LiWi-Crello	**0.0512**	**0.0423**	**0.0345**	**0.0331**	**0.0323**	**0.0321**	**0.8823**	**0.8956**	**0.9224**	**0.9250**	**0.9279**	**0.9308**

Zero-Shot Foreground Segmentation. To further assess predicted alpha masks, we evaluate foreground segmentation on DIS-5K (Qin et al., 2022), a high-resolution real-world benchmark with fine structures and diverse objects. As shown in Table 3, LiWi produces foreground masks competitive with specialized segmentation methods, despite having never seen these data. This suggests that LiWi framework possesses strong zero-shot foreground segmentation capabilities..

Table 3: Comparison of various methods on the foreground segmentation.

Methods	TE (1-4)					VD				
	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$
BASNet (Fan et al., 2020)	.744	.664	.092	.786	.814	.737	.656	.094	.781	.809
U^2Net (Qin et al., 2020)	.771	.676	.082	.799	.825	.753	.656	.089	.785	.809
HRNet (Wang et al., 2020)	.743	.658	.087	.781	.840	.726	.641	.095	.767	.824
PGNet (Xie et al., 2022)	.809	.746	.063	.830	.885	.798	.733	.067	.824	.879
IS-Net (Qin et al., 2022)	.799	.726	.070	.819	.858	.791	.717	.074	.813	.856
FP-DIS (Zhou et al., 2023)	.831	.770	.047	.847	.895	.823	.763	.062	.843	.891
BiRefNet (Zheng et al., 2024b)	.896	.858	.035	.901	.934	.891	.854	.038	.898	.931
LiWi (Zero-shot)	.794	.744	.077	.821	.871	.791	.744	.078	.821	.871

5.3 Qualitative Results

Qualitative Layer Decomposition. Qualitative comparisons are presented in Fig. 7. LiWi consistently yields more faithful layered decompositions for in-the-wild images. In the plant example, Qwen-Image-Layered only extracts partial leaves, while the SFT variant introduces floating branch artifacts and erroneously removes background curtain structures. In contrast, LiWi preserves the foreground plant while maintaining background integrity. Furthermore, in indoor scenes where baselines leave residual contact shadows or dark regions indicating incomplete foreground-background disentanglement, LiWi effectively eliminates these artifacts to produce cleaner results.

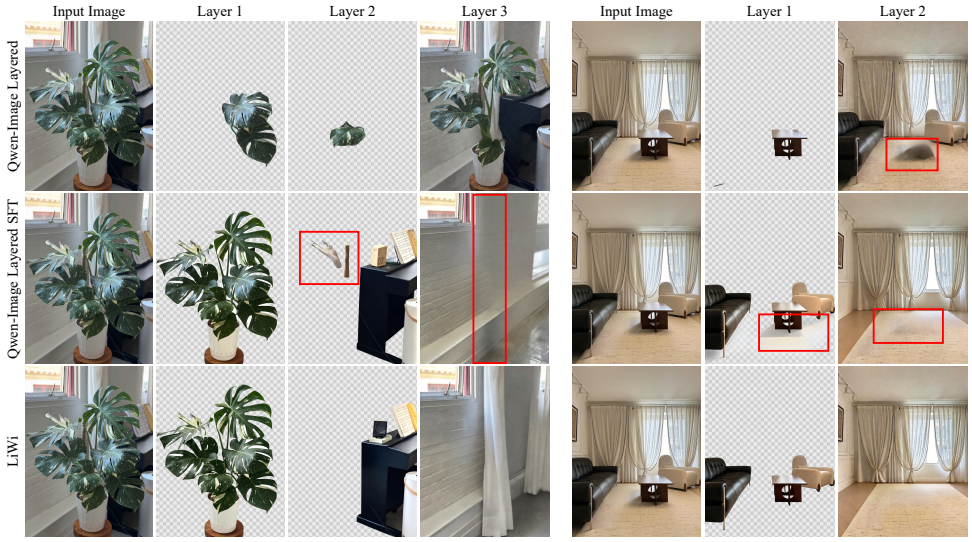

Figure 7: Qualitative comparison on in-the-wild layer decomposition.

Table 4: Ablation of reconstruction targets and degradation-restoration objective on LiWi-100k.

Metric	RGB L1↓						Alpha soft IoU↑					
# Max Edits	0	1	2	3	4	5	0	1	2	3	4	5
Qwen-Image-Layered-SFT	0.0996	0.0993	0.0974	0.0963	0.0953	0.0946	0.8745	0.8754	0.8783	0.8801	0.8816	0.8828
- Source Image Reconstruction	0.0950	0.0951	0.0944	0.0939	0.0935	0.0932	0.9177	0.9199	0.9216	0.9229	0.9239	0.9247
+ Latent-space Photometric Residual	0.0971	0.0971	0.0965	0.0961	0.0958	0.0956	0.9169	0.9178	0.9203	0.9218	0.9230	0.9239
+ Pixel-space Photometric Residual	0.0910	0.0912	0.0904	0.0897	0.0893	0.0890	0.9243	0.9246	0.9268	0.9281	0.9292	0.9300
+ Degradation-restoration Objective	**0.0907**	**0.0906**	**0.0895**	**0.0889**	**0.0883**	**0.0879**	**0.9314**	**0.9319**	**0.9332**	**0.9340**	**0.9347**	**0.9353**

5.4 Ablation Study

We ablate the reconstruction targets in the layered diffusion objective in Table 4. Qwen-Image-Layered-SFT reconstructs the source image, creating a shortcut and allowing the model to bypass layer-wise reasoning. Removing this objective (- source image reconstruction) substantially improves RGB L1 and Alpha IoU, indicating that direct image reconstruction weakens layer-level supervision.

We further investigate photometric residual supervision in latent and pixel spaces. The latent-space variant computes the residual after encoding, while the pixel-space variant computes it in image space. The pixel-space variant performs better, since photometric residual in latent space tend to be less expressive. Compared with removing source reconstruction, pixel-space residual prediction further reduces RGB L1 by 4.34% and improves alpha soft IoU by 0.58% on average. This suggests that explicitly modeling photometric residuals helps the model better handle illumination-induced errors.

Finally, adding the degradation-restoration objective further improves alpha quality. It improves alpha soft IoU by 0.67% on average over pixel-space residual prediction, while yielding a smaller RGB L1 gain of 0.87%, consistent with its focus on refining boundary defects.

6 Conclusion

We presented LiWi, a framework for decomposing in-the-wild images. To enable scalable supervision, we introduced ADD that automatically constructs in-the-wild layered images, resulting in LiWi-100k. We further propose a novel framework for natural image layering. The photometric residual layer captures illumination variations. Meanwhile, the degradation-restoration objective provides auxiliary boundary-correction supervision. Extensive experiments demonstrate that LiWi not only improves both RGB fidelity and alpha accuracy but also facilitates strong zero-shot foreground segmentation.

AI USE STATEMENT

Generative AI tools were used in the ADD data-construction pipeline: Qwen3-VL-32B generated removal instructions and verified decomposition proposals, and FLUX.2-klein-9B performed image editing to construct background and foreground content, as described in the data-construction section. Generative AI tools were also used to polish and translate the manuscript and to assist with simple utility code and tests. For AI-assisted writing and code, the authors inspected the outputs and incorporated them into the manuscript or Git repository only after checking them and finding no issues. Apart from the data-construction uses described above, research design, interpretation of results, and mathematical reasoning and proofs were conducted by the authors without generative-AI assistance. The authors take responsibility for the final text, code, data, claims, and results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, pp. 707–723, 2022.

Black Forest Labs. FLUX.2-klein-9B, 2026. URL https://huggingface.co/black-forest-labs/FLUX.2-klein-9B. Official model card. Accessed: 2026-09-25.

BRIA AI. Rmbg-1.4: Background removal model, 2024a. URL https://huggingface.co/briaai/RMBG-1.4. Accessed: 2026-04-27.

BRIA AI. Rmbg-2.0: Background removal model, 2024b. URL https://huggingface.co/briaai/RMBG-2.0. Accessed: 2026-04-27.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment Anything with Concepts. *arXiv preprint arXiv:2511.16719*, 2025. URL https://arxiv.org/abs/2511.16719.

Fangyi Chen, Yaojie Shen, Lu Xu, Ye Yuan, Shu Zhang, Yulei Niu, and Longyin Wen. Referring layer decomposition. *arXiv preprint arXiv:2602.19358*, 2026.

Jingxi Chen, Yixiao Zhang, Xiaoye Qian, Zongxia Li, Cornelia Fermuller, Caren Chen, and Yiannis Aloimonos. From inpainting to layer decomposition: Repurposing generative inpainting models for image layer decomposition. *arXiv preprint arXiv:2511.20996*, 2025a.

Jingye Chen, Zhaowen Wang, Nanxuan Zhao, Li Zhang, Difan Liu, Jimei Yang, and Qifeng Chen. Rethinking layered graphic design generation with a top-down approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16861–16870, 2025b.

Yusuf Dalva, Yijun Li, Qing Liu, Nanxuan Zhao, Jianming Zhang, Zhe Lin, and Pinar Yanardag. Layerfusion: Harmonized multi-layer text-to-image generation with generative priors. In *NeurIPS 2025 Workshop on Space in Vision, Language, and Embodied AI*, 2024.

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2777–2787, 2020.

Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. A survey on intrinsic images: Delving deep into lambert and beyond. *International Journal of Computer Vision*, 130: 836–868, 2022.

Dingbang Huang, Wenbo Li, Yifei Zhao, Xinyu Pan, Yanhong Zeng, and Bo Dai. Psdiffusion: Harmonized multi-layer image generation via layout and appearance alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3233–3242, 2026.

Jiahui Huang, Jun Gao, Vignesh Ganapathi-Subramanian, Hao Su, Yin Liu, Chengcheng Tang, and Leonidas J Guibas. Deepprimitive: Image decomposition by layered primitive detection. *Computational Visual Media*, 4(4):385–397, 2018.

Junjia Huang, Pengxiang Yan, Jinhang Cai, Jiyang Liu, Zhao Wang, Yitong Wang, Xinglong Wu, and Guanbin Li. Dreamlayer: Simultaneous multi-layer generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3357–3366, 2025.

Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. In *European Conference on Computer Vision*, pp. 144–160. Springer, 2024.

Kyoungkook Kang, Gyujin Sim, Geonung Kim, Donguk Kim, Seungho Nam, and Sunghyun Cho. Layeringdiff: Layered image synthesis via generation, then disassembly with generative knowledge. *arXiv preprint arXiv:2501.01197*, 2025.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics*, 40(6):1–12, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14317–14326, 2023.

Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting Anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1775–1785, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024W/MMFM/html/Li_Matting_Anything_CVPRW_2024_paper.html.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Cheng Liu, Yiren Song, Haofan Wang, and Mike Zheng Shou. Omnipsd: Layered psd generation with diffusion transformer. *arXiv preprint arXiv:2512.09247*, 2025.

M. Oquab et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7952–7962, 2025.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. 106:107404, 2020.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, et al. Highly accurate dichotomous image segmentation. 2022.

Haoran Sun, Haoyu Bian, Shaoning Zeng, Yunbo Rao, Xu Xu, Lin Mei, and Jianping Gou. Dataseta-gent: A novel multi-agent system for auto-constructing datasets from real-world images. *arXiv preprint arXiv:2507.08648*, 2025.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022.

Tomoyuki Suzuki, Kang-Jun Liu, Naoto Inoue, and Kota Yamaguchi. Layerd: Decomposing raster graphic designs into layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17783–17792, 2025.

Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22413–22422, 2024.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. 43(10):3349–3364, 2020.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image Technical Report. *arXiv preprint arXiv:2508.02324*, 2025. URL https://arxiv.org/abs/2508.02324.

Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. 2022.

Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5481–5489, 2021.

Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Generative image layer decomposition with visual effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7643–7653, 2025.

Jinrui Yang, Qing Liu, Yijun Li, Mengwei Ren, Letian Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Controllable layered image generation for real-world editing. *arXiv preprint arXiv:2601.15507*, 2026.

Shengming Yin, Zekai Zhang, Zecheng Tang, Kaiyuan Gao, Xiao Xu, Kun Yan, Jiahao Li, Yilei Chen, Yuxiang Chen, Heung-Yeung Shum, et al. Qwen-image-layered: Towards inherent editability via layer decomposition. *arXiv preprint arXiv:2512.15603*, 2025.

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024.

Xinyang Zhang, Wentian Zhao, Xin Lu, and Jeff Chien. Text2layer: Layered image generation using latent diffusion model. *arXiv preprint arXiv:2307.09781*, 2023.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 2024a.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *arXiv preprint arXiv:2401.03407*, 2024b.

Yan Zhou, Bo Dong, Yuanfeng Wu, Wentao Zhu, Geng Chen, and Yanning Zhang. Dichotomous image segmentation with frequency priors. 2023.

A Human Audit of Layered Data

Three annotators independently evaluated 1,000 randomly sampled examples from the original LiWi-100k release and 1,000 candidates before verifier filtering. They assessed three criteria: artifact-free layers, alpha-boundary quality, and recomposition consistency. An example received a final pass only when it passed all three criteria for that annotator. Table 5 reports per-annotator pass rates and their averages; the average final pass rate is not a majority-vote acceptance rate. The audit is used only to assess data quality and does not introduce manual annotation into the construction pipeline.

Table 5: Human audit of 2,000 examples: 1,000 original LiWi-100k samples and 1,000 candidates before verifier filtering. All values are pass rates (%). Final pass requires satisfying all three criteria.

Data setting	Annotator	Artifact-free layers	Alpha-boundary quality	Recomposition consistency	Final pass
LiWi-100k	1	96.9	97.1	97.1	95.0
	2	96.5	96.3	97.5	94.4
	3	96.6	96.7	98.1	95.4
	Average	**96.7**	**96.7**	**97.6**	**94.9**
w/o Verifier	1	83.9	86.9	62.7	52.7
	2	83.2	85.2	62.4	51.6
	3	83.4	86.0	63.8	53.9
	Average	83.5	86.0	63.0	52.7

The average final pass rate increases from 52.7% before verifier filtering to 94.9% in LiWi-100k, supporting the verifier's role in selecting visually plausible and consistent samples.

B Evaluation Protocol and Metrics

We follow the layer-sequence evaluation protocol of LayerD (Suzuki et al., 2025), also adopted by Qwen-Image-Layered (Yin et al., 2025). The definitions below specify the conventions in LayerD's public evaluation implementation.[1] Let $\hat{\mathcal{Y}} = (\hat{L}_1, \ldots, \hat{L}_N)$ and $\mathcal{Y} = (L_1, \ldots, L_M)$ denote predicted and reference RGBA layers ordered from bottom to top. Each layer $L_j = (C_j, \alpha_j)$ has RGB values C_j and opacity α_j on a common pixel grid Ω; hats indicate predictions. RGB and alpha values are normalized to $[0, 1]$ by dividing their 8-bit values by 255.

RGB L1. For a predicted-reference pair $(\hat{L}_i, L_j)$, define the reference support mask $w_j(p) = \mathbf{1}[\alpha_j(p) > 0]$. The RGB error is

$$d_{\mathrm{RGB}}(\hat{L}_i, L_j) = \frac{\sum_{p \in \Omega} w_j(p) \left(\frac{1}{3} \sum_{c=1}^{3} |\hat{C}_{i,c}(p) - C_{j,c}(p)| \right)}{\sum_{p \in \Omega} w_j(p) + \varepsilon}, \qquad \varepsilon = 10^{-8}. \tag{4}$$

Thus, RGB differences are averaged equally over pixels with nonzero reference alpha, including semitransparent pixels; fully transparent reference pixels are excluded. This specifies the binary-support weighting used by the reference code's `RGBL1` class. The metric evaluates individual layer appearances, including content occluded in the flattened image when it is present in the reference layer. Lower values indicate better RGB fidelity.

[1] `https://github.com/CyberAgentAILab/LayerD/tree/main/src/layerd/evaluation`

Alpha soft IoU. The reference implementation uses the fuzzy intersection and union of the continuous alpha maps:

$$s_\alpha(\hat{L}_i, L_j) = \frac{\sum_{p\in\Omega} \min(\hat{\alpha}_i(p), \alpha_j(p))}{\sum_{p\in\Omega} \max(\hat{\alpha}_i(p), \alpha_j(p))}. \tag{5}$$

If the denominator is zero, the implementation returns zero. No thresholding is applied to the alpha maps for this metric, so fractional opacity and soft boundaries contribute to the score. Higher values indicate better alpha agreement.

Layer grouping and alignment. The evaluator first composites layers that are unoccluded by higher layers into a top-layer group, removes that group, and repeats on the remaining layers. This visibility-based preprocessing is applied to both sequences and preserves their rendered appearance. The resulting groups are ordered from bottom to top and aligned using dynamic time warping (DTW). The pairwise alignment cost is

$$d_{ij} = \frac{d_{\mathrm{RGB}}(\hat{L}_i, L_j) + 1 - s_\alpha(\hat{L}_i, L_j)}{2}. \tag{6}$$

For the grouped sequences, the accumulated cost satisfies

$$D_{ij} = d_{ij} + \min\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\}, \tag{7}$$

with $D_{00} = 0$ and $D_{i0} = D_{0j} = +\infty$ for positive indices. Backtracking from the last pair yields an order-preserving path P that covers both sequences and allows a layer to match multiple layers. Consequently, different layer counts do not require an arbitrary one-to-one assignment.

Max Edits. Max Edits E is the total budget for additional greedy merge operations after the initial grouping. One edit composites two adjacent layers in either the predicted or reference sequence into a single RGBA layer, preserving their stacking order. For lower layer L_a and upper layer L_b, alpha compositing gives $\alpha_{ab} = \alpha_b + (1-\alpha_b)\alpha_a$ and $\alpha_{ab}C_{ab} = \alpha_b C_b + (1-\alpha_b)\alpha_a C_a$. At each iteration, the evaluator selects the candidate with the largest positive local matching gain under LayerD's merge criterion, applies it, and recomputes DTW. Merges cannot reduce either sequence below two layers. The procedure stops once E edits have been used or no valid improving merge remains. If it stops early, its last scores are retained for larger budgets. At $E = 0$, the initial grouping and DTW still apply, but no additional merge is performed. Thus, increasing Max Edits relaxes the required decomposition granularity while keeping the model predictions fixed; it does not change the number of inference steps.

Aggregation. Let $P_n^{(E)}$ be the final alignment for test image n under budget E, with T test images. For either pairwise metric $q \in \{d_{\mathrm{RGB}}, s_\alpha\}$, the reported score is

$$\mathrm{Score}_q(E) = \frac{1}{T}\sum_{n=1}^{T} \frac{1}{|P_n^{(E)}|} \sum_{(i,j)\in P_n^{(E)}} q\Big(\hat{L}_{n,i}^{(E)}, L_{n,j}^{(E)}\Big), \tag{8}$$

where the superscript denotes layers after the permitted merges. We first average over aligned pairs within each image and then average over test images, giving each image equal weight.

C Visual Prompt for Layer Decomposition.

To improve the generalizability and controllability of our method, we introduce visual-prompt-based layer decomposition. As shown in Fig. 8, given a user-specified bounding box that indicates the region to be separated, our model decomposes the corresponding content into an editable layer with an alpha mask. This process can be applied iteratively, enabling users to progressively decompose multiple regions in complex scenes.



Figure 8: Layer decomposition guided by visual prompt.

D Complete Category Distribution of LiWi-100k

Table 6 reports the complete category distribution of LiWi-100k, with 16 fine-grained categories grouped into six broad categories. All percentages are measured over the full dataset, and each broad-category share is the sum of its constituent fine-grained shares.

Table 6: Complete broad and fine-grained category distribution of LiWi-100k.

Broad category	Percentage	Fine-grained category	Percentage
Lifestyle	34.1%	Food & Dining Scenes	14.7%
		Human-Centered Scenes	11.9%
		Apparel & Accessories	4.8%
		Sports, Toys & Recreation	2.7%
Home	27.5%	Household Objects & Tools	11.3%
		Furniture & Home Décor	9.0%
		Indoor Rooms & Workspaces	7.2%
Nature	14.1%	Plants & Flowers	5.6%
		Animals & Pets	4.3%
		Natural Landscapes & Waterscapes	4.2%
Products	11.9%	Consumer Products & Packaging	7.1%
		Electronics & Appliances	4.8%
Urban	9.1%	Architecture & Street Scenes	5.2%
		Vehicles & Transportation	3.9%
Visuals	3.3%	Artwork, Text & Signage	2.2%
		Others	1.1%

E Computational Cost of ADD and LiWi

We distinguish offline data construction, model training, and inference. BIC, FIC, and LIC are used only during offline data construction to generate and filter layer tuples; different source images can be processed independently in parallel. LiWi training directly loads the cached layer tuples and does not invoke ADD. The degradation-restoration objective is used only during training. Neither ADD nor degradation-restoration is used during inference.

Table 7 reports wall-clock time, throughput, and inference computation. Training uses 16 NVIDIA B200 GPUs, a batch size of 16, and 12K optimization steps. Both inference benchmarks use one NVIDIA B200 GPU, two layers, a resolution of 1024×1024, and 50 sampling steps over 32 images.

Under this setting, LiWi has the same inference cost as Qwen-Image-Layered. The multi-model overhead of ADD is incurred only during offline construction; researchers using the released dataset do not need to rerun ADD.

Table 7: Computational cost of ADD and LiWi. Time is wall-clock time in hours; inference time is measured over 32 images. All GPUs are NVIDIA B200.

Stage	GPUs	Time (h)	Throughput / inference computation
BIC and FIC	8	88	2.5K source images/h
LIC	8	85	10.9K candidates/h
LiWi training	16	39	308 steps/h
Qwen-Image-Layered inference	1	0.8	40 images/h 242 TFLOPs/image
LiWi inference	1	0.8	40 images/h 242 TFLOPs/image

F Comparison of ADD with a MuLAn-style Pipeline

MuLAn (Tudosiu et al., 2024) sequentially performs instance discovery, occlusion ordering, instance completion, and reassembly. Because its data-construction code was unavailable, we implemented a MuLAn-style pipeline from the paper. We could not reproduce MuLAn's two failure classifiers trained on 5K manual annotations, so this comparison does not include the complete MuLAn filtering pipeline. On 500 shared source images, one annotator evaluated 100 randomly sampled final outputs per pipeline using the protocol in Appendix A. Final pass requires satisfying all three criteria: artifact-free layers, alpha-boundary quality, and recomposition consistency.

Table 8: Comparison of ADD with a MuLAn-style pipeline. All values are human-audited pass rates (%).

Pipeline	Artifact-free layers ↑	Alpha-boundary quality ↑	Recomposition consistency ↑	Final pass ↑
MuLAn-style pipeline	37	38	41	31
ADD	**95**	**95**	**96**	**93**

ADD achieves a final pass rate of 93%, compared with 31% for the MuLAn-style pipeline (Table 8). ADD decouples layer construction, forms valid layer combinations, and verifies them, reducing error propagation across stages.

G Sensitivity to the VLM Backbone

To isolate VLM sensitivity, we replaced the agent on 500 source images, removed the verifier for all variants, and kept all other components fixed. One annotator evaluated 100 randomly sampled final outputs per variant using the protocol in Appendix A. Table 9 reports the pass rates. ADD remains functional with different VLMs, but the quality of generated candidates depends on the VLM: the final pass rate drops from 51% with Qwen3-VL-32B to 47% with Qwen3.5-27B and 40% with InternVL3.5-38B.

H Use of DINOv2 Features in the LIC

DINOv2 (Oquab et al., 2024) provides self-supervised visual features validated on image- and pixel-level tasks. The LIC uses these RGB features to check whether masked regions contain repeated

Table 9: Sensitivity to the VLM backbone with the verifier disabled. All values are human-audited pass rates (%).

VLM	Artifact-free layers ↑	Alpha-boundary quality ↑	Recomposition consistency ↑	Final pass ↑
InternVL3.5-38B	46	47	46	40
Qwen3.5-27B	52	51	53	47
Qwen3-VL-32B	**55**	**55**	**56**	**51**

entities and whether the recomposed image is consistent with the source image. It does not use DINOv2 to assess transparency or infer occlusion order.

To compare with pixel-wise distance, we replace DINOv2 cosine similarity in the Proposal Selector with $1 - \mathrm{MSE}$ on 500 layer combinations. One annotator evaluated 100 randomly sampled Proposal Selector outputs per variant using the protocol in Appendix A. As shown in Table 10, DINOv2 raises the final pass rate from 43% to 51%, supporting the use of its RGB features for filtering candidates under editing and inpainting variations.

Table 10: DINOv2 cosine similarity versus pixel-wise distance in the Proposal Selector. All values are human-audited pass rates (%).

Selector similarity	Artifact-free layers ↑	Alpha-boundary quality ↑	Recomposition consistency ↑	Final pass ↑
$1 - \mathrm{MSE}$	51	47	49	43
DINOv2 cosine similarity	**55**	**55**	**56**	**51**

I Additional visualization Results

I.1 Additional visualization Results generated on LiWi framework

To further demonstrate the generative capabilities of our proposed framework, we provide additional image samples generated on the test set of LiWi-100k. It can be seen in Fig. 9 that our method can accomplish layered tasks in diverse scenarios, including simultaneous layering of multiple objects, portrait layering, layering under specific lighting conditions, etc. While completing layering with high quality, our method maintains the consistency of lighting and shadow.

I.2 Additional visualization Results on LiWi-100k dataset

To further demonstrate the diversity of the dataset, we present additional image samples from LiWi-100k. As shown in the Fig. 10 and Fig. 11, across both diverse scene categories and semantically rich multi-level decomposition tasks, our dataset construction method can consistently generate high-quality hierarchical results with high accuracy. While performing semantic decomposition, it also preserves the consistency and richness of lighting and shadow effects.

J Complete Zero-Shot Foreground Segmentation Results on Real-World Data

We compare our method with various foreground segmentation approaches on the four test sets and the validation set of DIS5K. The results show that our method achieves competitive performance under the zero-shot setting.



Figure 9: Results of LiWi framework on the test set of LiWi-100k. For various natural scenes with multiple categories and diverse illumination conditions, our method can perform high-quality layering while maintaining the consistency of light and shadow.

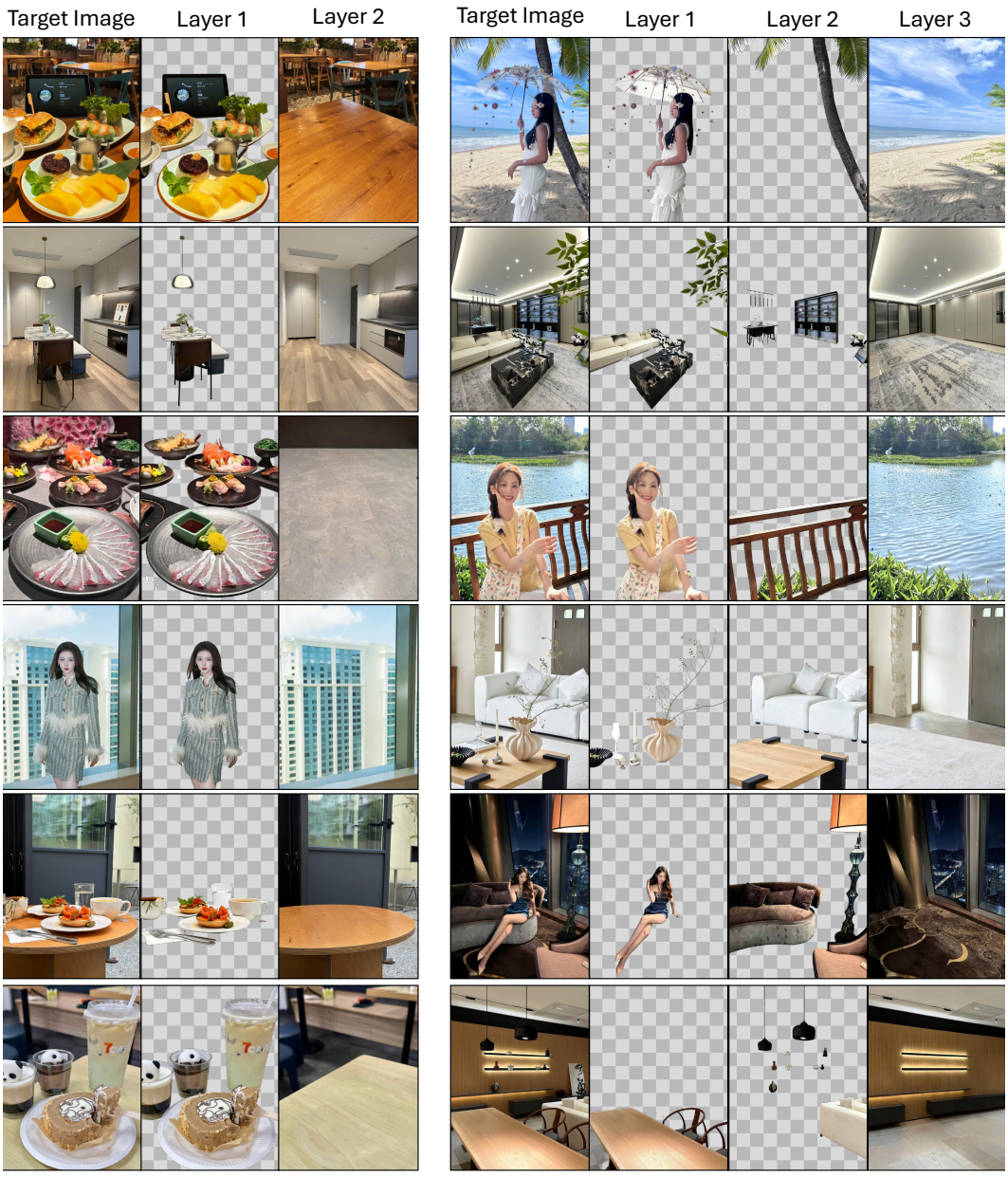

Figure 10: Visualization of the Liwi dataset with 2 and 3 layers. As shown, in diverse scenes, our construction method can generate high-quality layered images.

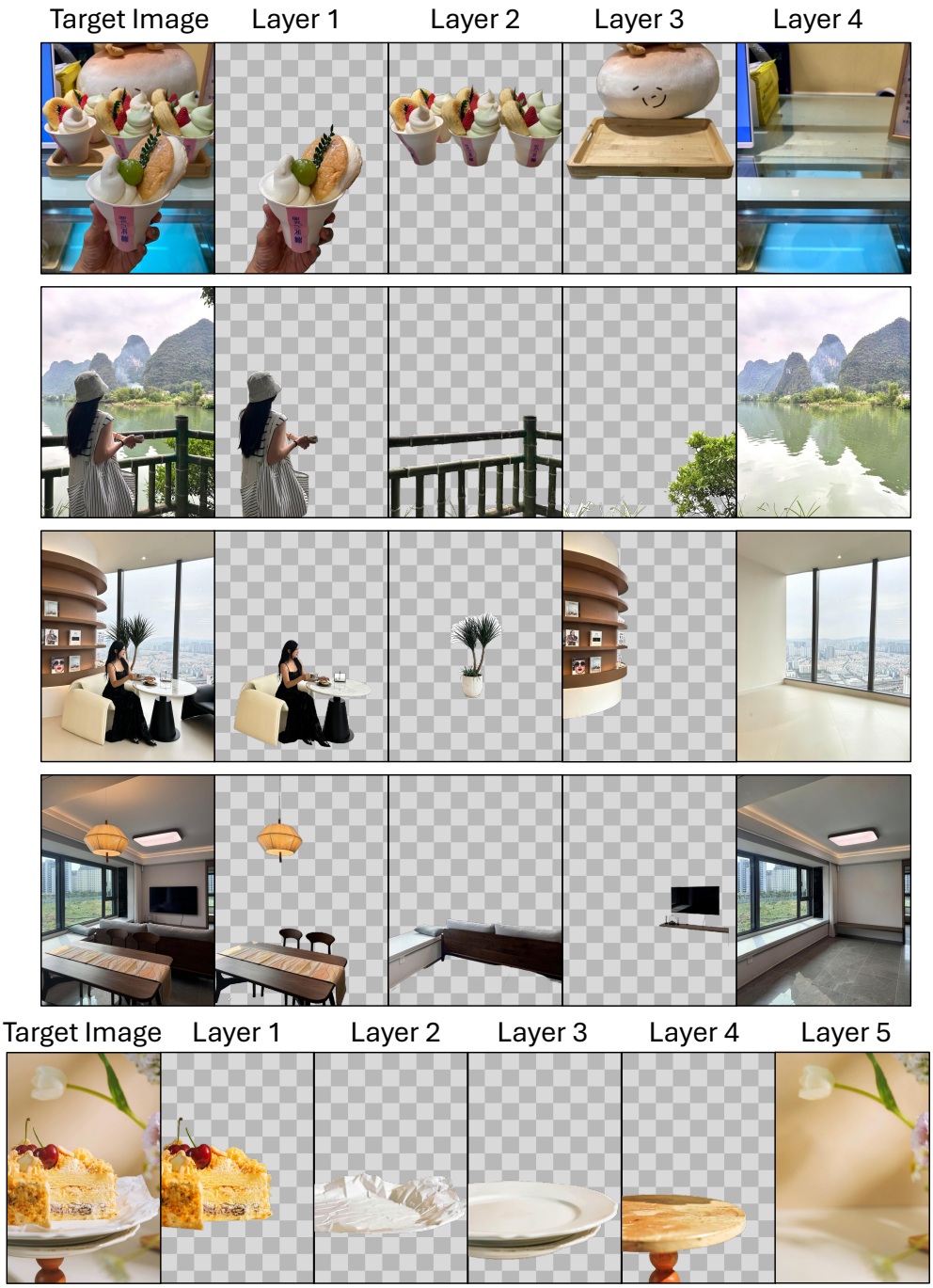

Figure 11: Visualization of the LiWi-100k dataset across multiple layers and aspect ratios. As the number of layers increases and the aspect ratio changes, our dataset construction method can still produce high-quality hierarchical results with semantic meaning.

Table 11: Comparison of various methods on the foreground segmentation task. In the zero-shot setting, our method achieves performance close to that of dedicated foreground segmentation models.

Methods	TE1						TE2					
	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$
BASNet	.663	.577	.105	.741	.756	155	.738	.653	.096	.781	.808	341
U^2Net	.701	.601	.085	.762	.783	165	.768	.676	.083	.798	.825	367
HRNet	.668	.579	.088	.742	.797	262	.747	.664	.087	.784	.840	555
PGNet	.754	.680	.067	.800	.848	162	.807	.743	.065	.833	.880	375
IS-Net	.740	.662	.074	.787	.820	149	.799	.728	.070	.823	.858	340
FP-DIS	.784	.713	.060	.821	.860	160	.827	.767	.059	.845	.893	373
UDUN	.784	.720	.059	.817	.864	140	.829	.768	.058	.843	.886	325
BiRefNet	.860	.819	.037	.885	.911	106	.894	.857	.036	.900	.930	266
Ours	.812	.755	.064	.836	.880	204	.825	.777	.065	.846	.891	505

Methods	TE3						TE4					
	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$
BASNet	.790	.714	.080	.816	.848	681	.785	.713	.087	.806	.844	2852
U^2Net	.813	.721	.073	.823	.856	738	.800	.707	.085	.814	.837	2898
HRNet	.784	.700	.080	.805	.869	1049	.772	.687	.092	.792	.854	3864
PGNet	.843	.785	.056	.844	.911	797	.831	.774	.065	.841	.899	3361
IS-Net	.830	.758	.064	.836	.883	687	.827	.753	.072	.830	.870	2888
FP-DIS	.868	.811	.049	.871	.922	780	.846	.788	.061	.852	.906	3347
UDUN	.865	.809	.050	.865	.917	658	.846	.792	.059	.849	.901	2785
BiRefNet	.925	.893	.028	.919	.955	569	.904	.864	.039	.900	.939	2723
Ours	.809	.760	.068	.833	.887	1026	.727	.685	.108	.767	.822	3700

Methods	TE (1-4)						VD					
	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$	$F_\beta^x \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_m \uparrow$	$E_\phi^m \uparrow$	$\text{HCE}_\gamma \downarrow$
BASNet	.744	.664	.092	.786	.814	1007	.737	.656	.094	.781	.809	1132
U^2Net	.771	.676	.082	.799	.825	1042	.753	.656	.089	.785	.809	1139
HRNet	.743	.658	.087	.781	.840	1432	.726	.641	.095	.767	.824	1560
PGNet	.809	.746	.063	.830	.885	1173	.798	.733	.067	.824	.879	1326
IS-Net	.799	.726	.070	.819	.858	1016	.791	.717	.074	.813	.856	1116
FP-DIS	.831	.770	.047	.847	.895	1165	.823	.763	.062	.843	.891	1309
UDUN	.831	.772	.057	.844	.892	977	.823	.763	.059	.838	.892	1097
BiRefNet	.896	.858	.035	.901	.934	916	.891	.854	.038	.898	.931	989
Ours	.794	.744	.077	.821	.871	1359	.791	.744	.078	.821	.871	1417

K Reproducibility Statement

We provide the ADD construction procedure and Proposal Selector in Section 3 and Algorithm 1, the model formulation and training objectives in Section 4, implementation details and data splits in Section 5.1, the complete evaluation protocol in Appendix B, and computational costs in Appendix E. We will release the code, evaluation scripts, and LiWi-100k with documentation needed to reproduce the reported results.