MEMO: Human-like Crisp Edge Detection Using Masked Edge Prediction

Supplementary Material

A. Network Details

The masked edge encoder and edge decoder have a symmetric architecture composed of four residual blocks [9] with 128, 256, 512, and 768 output channels, respectively. Each residual block consists of two convolutional sub-blocks. In each convolutional sub-block, the input feature is processed by a group normalization layer [29], a SiLU activation [5], and a 3×3 convolutional layer.

Overall, MEMO contains 238M parameters during pretraining, of which 86M belong to the DINOv2 [21] image encoder. After inserting the LoRA adapters [13], the total number of parameters increases to 241M, corresponding to only a 1.2% increase.

B. Flexibility of Granularity Scale

In Section 2.4, we discussed how adjusting the granularity scale controls the richness of predicted edges to support multi-granularity edge prediction. Based on visual observations, we typically set the scale in the range of 1.0 to 2.0. However, MEMO's prediction mechanism supports a much wider range of scale values, offering greater flexibility than prior methods.

Previous work, such as MuGE [36] and SAUGE [18], enables granularity control using either paired supervision or multi-layer interpolation. However, both are limited to a pre-defined and fixed control range. In contrast, MEMO supports any positive scale value. As shown in Figure 8, setting the scale close to zero suppresses all edges, while increasing the scale progressively reveals more edges. When the scale is moderately increased (*e.g.*, $s = 1.4$ achieves C* or $s = 1.8$ achieves AC*), MEMO produces clean and complete edges. Pushing the scale further (*e.g.*, $s = 4.0$) continues to increase edge richness, but at the cost of more false positives and spurious structures. This illustrates that although MEMO is highly flexible, extreme values should be used cautiously depending on the application needs.

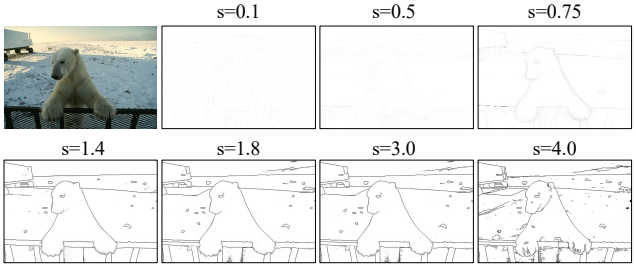

Figure 8. Edge predictions under varying granularity scale values. As the scale increases, more edges are progressively revealed. MEMO maintains high edge quality across a wide range of scales, but excessively large values may introduce spurious edges. This demonstrates MEMO's flexible granularity control beyond predefined ranges used in prior work.

C. Additional Discussion on Inference Steps

While Section 2.4 notes that MEMO typically finalizes most predictions within the first 10–20 steps, a trend also validated in the unmasking distribution shown in Figure 9, we find that the actual convergence speed is sample-dependent and strongly influenced by edge ambiguity. In particular, ambiguous edges that initially appear thick, fuzzy, or spatially uncertain tend to require more iterations to resolve. This behavior is illustrated in Figure 10. The second column shows MEMO's prediction after the first inference pass, which resembles outputs from existing methods: many edges are still thick and blurry. Yet these ambiguities are not uniform, some boundaries already appear relatively well-formed, while others remain diffuse and unstable.

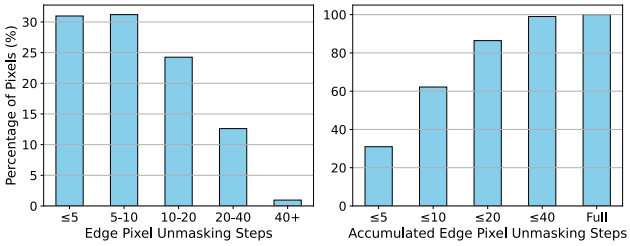

Figure 9. Distribution of edge pixel finalization steps across the inference process. Left: Histogram showing the percentage of edge pixels finalized at different unmasking step ranges. Right: Cumulative proportion of finalized pixels by step count.

As inference progresses, these edges converge at different speeds, as shown in the last two columns of Figure 10. Clear and confident boundaries typically stabilize within 20 steps as shown in green boxes, whereas ambiguous structures require substantially more iterations before reaching a consistent, thin contour as demonstrated in red boxes.

This observation suggests that the ideal number of inference steps should adapt to the visual complexity of the scene. It also explains why, as mentioned in Section 3.2, BIPED achieves visually crisp results with 5 steps, while it takes more steps for BSDS and Multicue to reveal sufficient crisp edges.

In Section 3.3, we observe that MEMO's performance

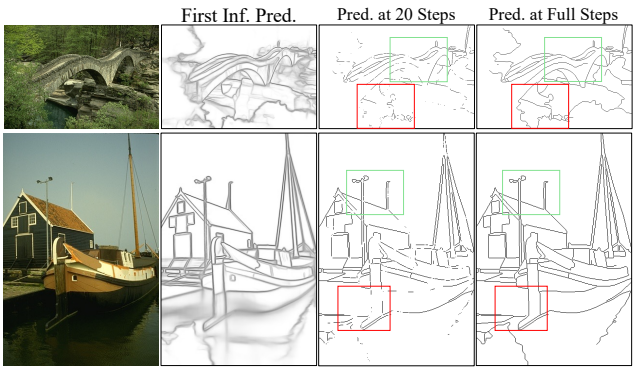

Figure 10. Intermediate predictions at different inference stages of MEMO. The second column shows edge predictions after the first inference pass, where all pixels are visualized. These predictions, similar to conventional methods, reveal thick and ambiguous prediction. However, the edges of different ambiguity converge at different speed. The last two columns show the finalized edge pixels after 20 steps and full inference, respectively. Less ambiguous boundaries usually converge early (green boxes), while highly uncertain regions (red boxes) require more iterations to stabilize.

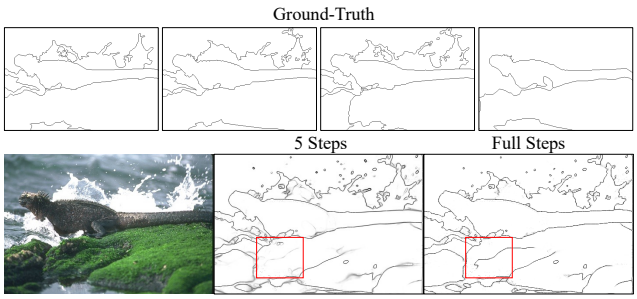

Figure 11. With fewer steps (5 steps), ambiguous edges are partially predicted with low confidence, resulting in faint contours. With full-step inference, MEMO finalizes these edges, leading to sharper but potentially unmatched predictions due to missing ground-truth labels. Although evaluation protocol may penalize these as false positives, they remain visually plausible from a human perspective.

under the SEval protocol slightly declines as the number of inference steps increases. However, this is not due to reduced prediction accuracy, but rather stems from the limitations of the evaluation metric. As illustrated in Figure 11, edges that are ambiguous typically receive lower confidence scores in early inference steps. As a result, when using a small number of steps (*e.g.*, 5 steps), these regions are often not finalized, leading to faint or partial predictions as shown in the red box. However, with more inference steps, MEMO is more likely to finalize these edges, especially if parts of them are progressively resolved with sufficient confidence. This behavior can cause perceptually valid but unannotated edges to be fully predicted, which are then incorrectly penalized as false positives. Despite the slightly lower SEval score at full-step inference, we argue that these predictions still reflect human visual perception and indicate strong prediction quality.

D. MEMO vs Edge Post-process

A common approach in edge detection pipelines is to rely on post-processing techniques, such as edge non-maximum suppression (NMS) or thinning to improve visual sharpness. These steps are often sufficient to boost benchmark scores under evaluation protocols, which permit a small spatial tolerance between predictions and ground-truth labels. However, high numerical scores do not necessarily imply high perceptual quality. In fact, these post-processed edges can diverge significantly from human-like annotations.

As shown in Figure 12, baseline methods [17, 18, 35, 36] that heavily depends on post-processing can produce spatially unstable edge predictions that are only superficially refined by thinning. For example, the antenna on the rooftop and the contours of the buildings frequently appear jagged, duplicated, or spatially misaligned even after post-processing. Such artifacts are tolerated by the evaluation metric but appear unnatural and imprecise to human observers. This becomes a critical limitation in applications where fine structure integrity or precise boundary localization is important.

MEMO addresses this issue from the root by incorporating crispness awareness directly into its prediction process. Rather than depending on external post-processing to refine ambiguous or overlapping predictions, MEMO progressively finalizes confident pixels while learning to suppress redundant activations in high-density regions. As a result, MEMO generates structurally coherent, well-localized, and perceptually aligned edge maps without any post-processing. This end-to-end design not only avoids the brittleness of hand-crafted post-processing rules but also leads to better generalization across different edge styles and levels of detail.

E. Example of Synthetic Datasets

When constructing the synthetic dataset, we follow the hyper-parameter settings recommended by SAM [15], with one modification: we set `stability_score_thresh` to 0.85. This increases the number of recalled objects and, in turn, provides a richer set of edge candidates for training.

Figures 13 and 14 show random samples of generated synthetic dataset with source images from LAION [23] and their edge side-by-side. Images are center cropped to 256×256 for better visualization, the actual source image and corresponding edge in the dataset has preserve their original aspect ratio.

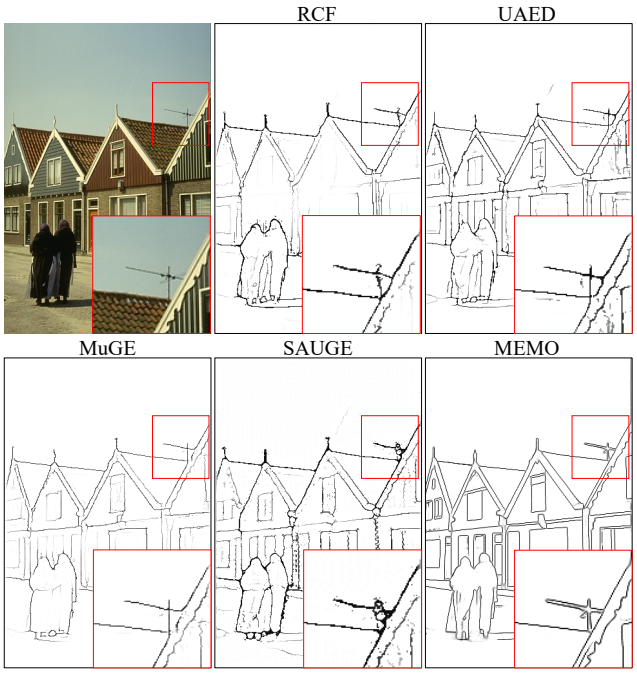

Figure 12. Qualitative comparison between MEMO without post-processing and baseline methods with post-processing. Although post-processing enhances benchmark scores, it does not guarantee perceptually faithful edge maps. Baseline predictions often exhibit duplicated, broken, or jittery edges, particularly in fine structures like the antenna and rooflines, due to the inherent ambiguity in thick or overlapping predictions. In contrast, MEMO directly predicts crisp and coherent edge structures, accurately capturing both global layouts and fine details without the need for post-processing.

F. Related Works

Edge Detection Deep learning-based edge detectors [8, 17, 22, 30, 32, 35, 36] typically formulate edge detection as a pixel-wise binary classification problem optimized with a binary cross-entropy loss. HED [30] introduces holistically-nested deep supervision with multi-scale side outputs to improve training stability and performance. RCF [17] aggregates richer convolutional features from multiple stages of the backbone to better capture both low-level and high-level cues. BDCN [8] adopts a bi-directional cascade structure with layer-specific supervision to learn scale-aware edge responses. EDTER [22] replaces purely convolutional backbones with a transformer-based architecture to exploit long-range dependencies for edge detection. DiffusionEdge [32] formulates edge detection as a conditional diffusion process, using a diffusion probabilistic model to iteratively predict edge maps. UAED [35] explicitly models annotation uncertainty by leveraging multiple annotations to learn an uncertainty-aware edge predictor. MuGE [36] extends this idea to multiple granularity levels, supervising edges at different granularity and strengths to better align with diverse annotation patterns. SAUGE [18] leverages the prior from segmentation models, training an adaption network to map the intermediate features of segmentation models to edge prediction. However, as long as the task is treated as binary classification on thick binary edge maps and optimized with cross-entropy loss, the predictions tend to suffer from the thick-edge issue, where the output is a band of responses around the true contour rather than a single-pixel-wide edge as produced by human annotators.

Crisp Edge Detection To mitigate the thick-edge problem, several methods have been proposed to encourage crisp, thin edge predictions. CED [33] adds side-refinement modules and supervision to sharpen boundary localization and suppress off-edge activations. LPCB [3] formulates boundary prediction with a loss design and training strategy that emphasize precise, localized responses within each patch, thereby improving contour crispness. CATS [14] analyzes and "unmixes" convolutional features, introducing feature unmixing and refinement modules to reduce blurry activations and produce sharper edges. Refined-label training [31] observes that label noise and misaligned annotations are a key source of thick predictions, and therefore proposes a guided label refinement strategy to obtain cleaner, sharper edge labels for supervision. DiffusionEdge [32], by leveraging a diffusion generative backbone, also contributes to crisp edge detection by generating results that resembles to visually crisp edges.

G. Limitations and Future Work

MEMO relies on recursive inference to refine edge predictions. While this iterative process is crucial for achieving high crispness, it also introduces non-trivial computational overhead, making the current model unsuitable for real-time edge detection on high-resolution images.

There are several promising directions to improve the efficiency of MEMO, such as distilling the iterative refinement process into a single (or few) feed-forward passes, designing adaptive step schedules and early-stopping criteria that dynamically decide how many refinement steps are needed, or combining MEMO with lightweight backbones and multi-scale tiling strategies for large images. However, each of these directions would require substantial algorithmic and system-level engineering, and a careful study of the trade-off between accuracy and efficiency. We therefore leave them as important directions for future work.

Figure 13. Example of synthetic edge dataset. Images are cropped to 256×256 for visualization.

Figure 14. Example of synthetic edge dataset. Images are cropped to 256×256 for visualization.