Reading the Unreadable: Text-Aware Image Super-Resolution Needs Reasoning

Jaeseong Lee **Jinwoo Kim** **Jinho Jeong** **Seon Joo Kim**
CIPLAB, Yonsei University

Project Page: https://jasonleex1995.github.io/TAISR_needs_Reasoning

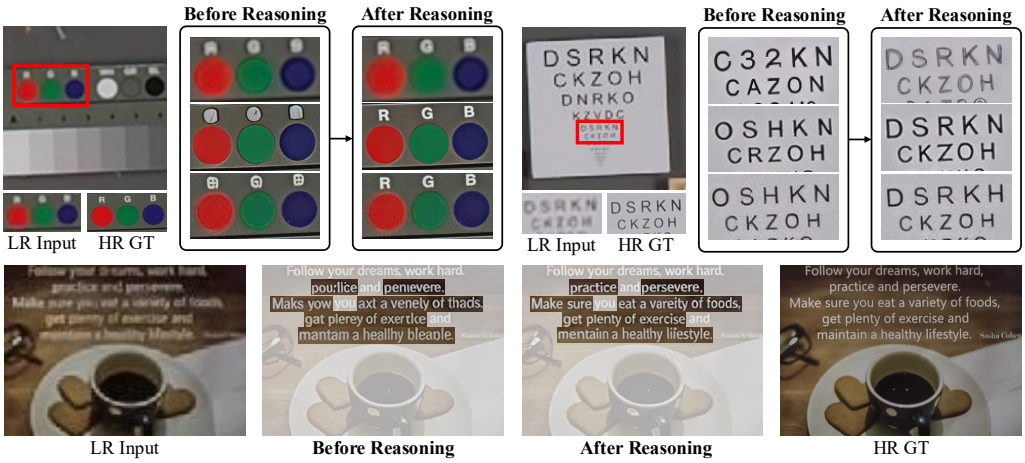


Figure 1: **TAISR needs reasoning.** Each panel shows the LR input, baseline SR outputs (Before Reasoning), outputs with RTC (After Reasoning), and the HR ground truth. The highlighted text is unreadable from local pixels alone, so the baselines hallucinate characters. With reasoning, the same models recover most of it.

Abstract

Text-aware image super-resolution (TAISR) aims to recover high-resolution images from low-resolution inputs while preserving textual content. Existing methods treat this as a visual restoration problem, yet severely degraded text often cannot be read from local visual evidence alone. Humans instead read it by reasoning over image context, logical patterns, and prior knowledge. This raises a fundamental question: *Does TAISR need reasoning?* To answer this, we introduce two diagnostic tools. The first, ReasonText, is a human-curated benchmark that separates reasoning-required text from locally recoverable text. The second, Ground-Truth Text-Crop Accuracy (GTTCA), is a recognition-based metric that measures text restoration quality without text detection errors. Evaluating twelve recent super-resolution (SR) models, we find a consistent and substantial performance drop on reasoning-required text, revealing a systematic blind spot. Two observations point to a remedy. One is that text-conditioned SR models restore text substantially better when given ground-truth text. The other is that multimodal large language models (MLLMs) can read degraded text in low-resolution images through reasoning. We therefore propose Reasoning Transfer via Captioning (RTC), a training-free plug-in that uses an external MLLM to infer the degraded text and feeds the resulting caption to a text-conditioned SR model. RTC consistently improves all six text-conditioned SR models we evaluate, especially on reasoning-required text. Surprisingly, a real-world SR model with RTC can even outperform the TAISR model with RTC, suggesting that reasoning should be a core design axis for future TAISR.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Real-world image super-resolution (real-world SR) [16, 33] aims to recover high-resolution (HR) images from low-resolution (LR) inputs corrupted by unknown degradations. Recent advances [5, 9, 17, 31, 35, 38, 41] leverage text-to-image generation models [10, 18, 24, 26] as generative backbones, producing visually plausible details even from severely degraded inputs. However, such generative priors often fail to preserve the structural information present in the LR image [5, 9]. This issue is particularly severe for text, where altering a single stroke can change its meaning entirely. To address this challenge, a line of work has introduced text-aware image super-resolution (TAISR) [13, 15, 23, 40], which restores full scene images while preserving textual content. Unlike scene text image super-resolution (STISR) [4, 32], which operates on cropped text patches, TAISR inherits the open-scene setting of real-world SR and targets text fidelity within full images.

Existing TAISR methods augment the SR model with text-aware modules such as text detection [40], segmentation [13], or spotting [15, 23]. Although these modules improve the localization and restoration of textual regions, they remain centered on local visual evidence. By contrast, humans read what appears unreadable. When a distant sign is nearly illegible, we can often still read it by reasoning from the surrounding context, logical patterns, or prior knowledge of brands, places, and vocabulary. Current TAISR pipelines lack explicit mechanisms for such reasoning and hallucinate characters where local evidence is insufficient (Figure 1). This gap raises the central question of this paper: *Does TAISR need reasoning?* Answering it requires measurement, yet current tools fall short in two ways. Existing benchmarks [23] do not distinguish reasoning-required text from locally recoverable text, and their evaluation protocols rely on text spotting models [20, 46], whose detection errors contaminate the measurement of text restoration quality.

We address both shortcomings with two diagnostic tools. First, we introduce ReasonText, a human-curated benchmark that assigns each text instance a perception-grounded difficulty level, separating reasoning-required text from locally recoverable text. Second, we propose Ground-Truth Text-Crop Accuracy (GTTCA), a text restoration metric that runs text recognition directly on ground-truth text regions of SR outputs, bypassing the text detection stage. Using these tools, we evaluate twelve recent real-world SR and TAISR models and find that every model's accuracy drops sharply on reasoning-required text: the lowest GTTCA on locally recoverable text across all models (45.7%) still exceeds the highest on reasoning-required text (24.5%). This failure occurs regardless of architecture, generative backbone, or training paradigm, and persists even in models explicitly designed for text.

These results answer our question: *TAISR does need reasoning.* What remains is how to supply it, and we start from two observations. First, text-conditioned SR models [5, 9, 15, 23, 37, 41] restore reasoning-required text substantially better when the ground-truth transcripts are provided in their captions. Second, modern multimodal large language models (MLLMs) [12, 29] can infer locally unreadable text in LR images through reasoning. Building on these observations, we propose Reasoning Transfer via Captioning (RTC). In RTC, an external MLLM infers the degraded text and writes it into a caption, which then replaces the default caption of a text-conditioned SR model. Since both models remain frozen, RTC is training-free and can be attached to any such model.

Empirically, RTC consistently improves both real-world SR and TAISR models on ReasonText, and these gains carry over to the existing TAISR benchmarks [23], SA-Text-test and RealText. Surprisingly, the strongest real-world SR model [9] with RTC outperforms the strongest TAISR model [15] with RTC, even though the latter builds on the same backbone [10] and adds text-aware modules. This result suggests that, once reasoning is supplied, current text-aware modules may bring no additional benefit. Reasoning should therefore be a core design axis for TAISR, and a natural next step is to build it into the text-aware modules themselves.

Our contributions are summarized as follows:

- We introduce ReasonText, the first TAISR benchmark for evaluating reasoning-required text, and GTTCA, a metric that measures text restoration quality free of text detection errors.
- We evaluate twelve recent real-world SR and TAISR models on ReasonText and show that every model's accuracy drops sharply on reasoning-required text.
- We propose RTC, a training-free plug-in that injects an external MLLM's reasoning into text-conditioned SR models through captions. Extensive experiments show that RTC consistently improves both real-world SR and TAISR models.

2 Related Work

Real-World Image Super-Resolution (Real-World SR). Real-world SR aims to recover HR images from LR inputs corrupted by diverse, unknown degradations. Early GAN-based methods [16, 33] improve perceptual quality with adversarial training [11], but often introduce visual artifacts. To avoid such artifacts and restore realistic details even under severe degradations, recent work instead builds on generative models with strong priors, including diffusion and autoregressive (AR) models. Multi-step diffusion-based methods [5, 9, 17, 31, 38, 41] adapt pretrained text-to-image diffusion models [10, 24, 26] to real-world SR with conditioning modules such as ControlNet-style adapters [44]. Few-step diffusion-based methods [34, 37, 43] reduce the inference cost of iterative denoising by starting the diffusion process from the LR image instead of Gaussian noise, or by distilling a multi-step model into a single step. AR-based methods [25, 35] adapt a multimodal AR model [18] with post-training techniques from large language models. However, such strong priors can also hallucinate plausible yet incorrect details. This failure mode is especially problematic for text, where even a minor deviation can alter the meaning.

Text-Aware Image Super-Resolution (TAISR). Motivated by this failure mode, TAISR aims to restore full scene images while explicitly preserving textual content. TextSR [40] first detects text regions with an OCR model, then restores the text with its own STISR model and the rest of the image with a real-world SR model [33]. TADiSR [13] adds a text segmentation decoder to predict text masks jointly with the restored image. TeReDiff [23] and UniT [15] use a text spotting module that takes diffusion features as input and predicts text to guide restoration. Despite these diverse designs, all of them remain centered on local visual evidence and do not consider the reasoning that humans use to read locally unreadable text.

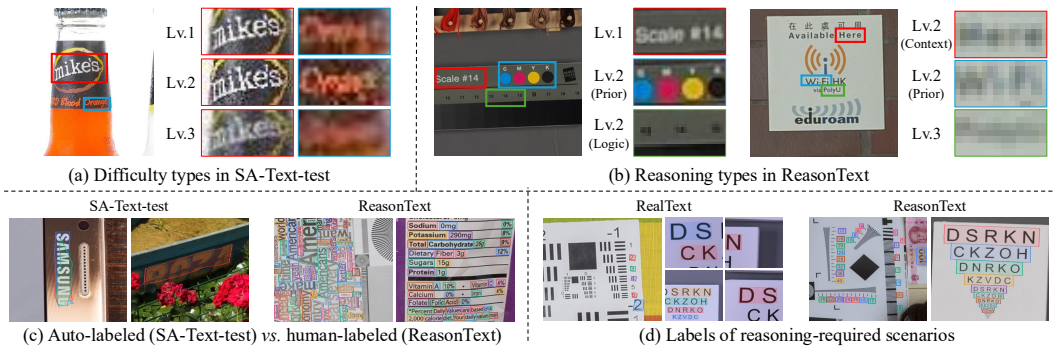
Multimodal Large Language Models (MLLMs). Recent MLLMs [1, 12, 28, 29] have demonstrated strong reasoning capabilities on benchmarks requiring joint visual and textual understanding [21, 42]. In image super-resolution, however, MLLMs have been used in a much more limited way. Real-world SR methods [5, 9, 41] employ LLaVA-1.5 [19] to caption LR images, and UniT adopts Qwen2.5-VL [2] to initialize text predictions. In both cases, the MLLM is an earlier-generation model with limited reasoning ability and serves only as a perceptual front-end. In contrast, we leverage the reasoning capability of recent MLLMs to infer locally unreadable text.

3 TAISR Needs Reasoning

In this section, we examine whether existing real-world SR and TAISR models can restore text that requires reasoning. In Section 3.1, we introduce ReasonText, a benchmark that separates reasoning-required text from locally recoverable text. In Section 3.2, we propose Ground-Truth Text-Crop Accuracy (GTTCA), a metric that measures text restoration quality by recognizing text directly at ground-truth locations. Using these two tools, we evaluate twelve recent SR models in Section 3.3 and find that all of them suffer a substantial accuracy drop on reasoning-required text.

3.1 ReasonText Benchmark

Limitations of existing benchmarks. To evaluate whether SR models can restore reasoning-required text, we need a *reasoning-required label* for each LR text instance, but no existing benchmark provides one. SA-Text-test [23] might seem to meet this need at first glance, since it provides three LR versions of each HR image at different synthetic degradation strengths. However, degradation strength is an *image-level* attribute, so it does not indicate whether an individual text instance requires reasoning (Figure 2(a)). A natural alternative is to add such labels to the existing benchmarks [23], SA-Text-test and RealText, but two of their characteristics rule this out. First, their text polygons and transcripts contain non-trivial errors, as they are produced by fully automatic pipelines without human verification (Figure 2(c)). Second, their images are cropped without considering the context of each text instance, leaving few instances that can be read through reasoning (Figure 2(d)). We therefore construct ReasonText with three key properties: (1) all text polygons and transcripts are annotated by humans, (2) each image is cropped to preserve the context of its text instances, and (3) each instance is labeled with a *perception-grounded* difficulty level: readable from the text region alone, readable only with reasoning, or unreadable (Figure 2(b)).



(a) Difficulty types in SA-Text-test

(b) Reasoning types in ReasonText

(c) Auto-labeled (SA-Text-test) *vs.* human-labeled (ReasonText)

(d) Labels of reasoning-required scenarios

Figure 2: **ReasonText addresses gaps in existing TAISR benchmarks.** (a) SA-Text-test assigns difficulty at the image level via three degradation strengths, which does not indicate whether each instance requires reasoning. (b) ReasonText instead labels each instance with a perception-grounded difficulty level, separating reasoning-required from locally recoverable text. (c) ReasonText is constructed from human annotations, which remain reliable across diverse scenes. (d) ReasonText crops the image to preserve context needed to read reasoning-required text.

Data collection and annotation. We build *ReasonText* on two real-world SR benchmarks, RealSR-v3 [3] and DRealSR [36], which provide camera-captured LR–HR pairs rather than synthetically degraded ones. Trained annotators first label word-level polygons, transcripts, and orientations for every alphanumeric text instance in each full HR image, and the labels go through multiple rounds of verification. They then crop a square HR patch around each text instance, including any content related to that instance. Following the standard $\times 4$ SR setting [15, 23], we resize each HR patch to 512×512 and its aligned LR patch to 128×128.

Difficulty labeling. We obtain the perception-grounded difficulty level of each text instance through a two-stage labeling process. Both stages follow the same protocol. Five of ten annotators are randomly assigned to each text instance, and transcribe the text only when they can read it, skipping it otherwise to prevent guessing. We then compare the collected transcripts with the ground-truth (GT) transcript, and regard a text instance as identified when they match.

Stage 1 (cropped LR text). We crop each text instance from its LR patch using the annotated polygon, yielding a *cropped LR text image* that contains no context outside the text itself. Instances identified at this stage are assigned **Level 1**. Since they are read from the text alone, we regard Level 1 instances as *locally recoverable text*.

Stage 2 (full LR patch). Stage 2 begins after Stage 1 is completed for all instances. We show each remaining instance again within its full LR patch, with its text region highlighted, so that annotators can use the surrounding context. When they can read the text, annotators also select which cue helped them most: *image context*, when the cue comes from what else is visible in the patch, such as the same text appearing in another region; *logical patterns*, when it comes from a regularity that the text follows, such as the numeric sequence on a ruler; or *prior knowledge*, when it comes from what the annotator already knows, such as a brand logo. Instances identified at this stage are assigned **Level 2**. Since they are read only once the surrounding context is available, we regard Level 2 instances as *reasoning-required text*. The rest are assigned **Level 3**. Since none of our annotators can read them even with the context, we treat Level 3 instances as *unrecoverable text*.

For Level 3 instances, an SR output that happens to match the GT transcript cannot be distinguished from a lucky guess. We therefore exclude Level 3 instances from evaluation and discard LR–HR pairs containing only Level 3 instances. The resulting benchmark contains 513 LR–HR pairs with 2,463 text instances (Level 1: 1,754, Level 2: 709). Further details are provided in Appendix A.1.

3.2 GTTCA Metric

Existing TAISR works [15, 23] evaluate text restoration quality using text spotting models [20, 46], which jointly perform text detection and recognition on the SR output. Detection errors therefore

Table 1: **Quantitative results of twelve real-world SR and TAISR models on ReasonText.** Every model struggles to recover Level 2 (reasoning-required) text. For each metric, the top-three models are in black (**best**, second-best, third-best plain) and the rest in gray. The bottom rows report Pearson/Spearman correlations between each metric and overall GTTCA across the twelve SR models. ↑ (↓): higher (lower) is better.

Group	Model	Fidelity		Perceptual					Text Restoration		
				Reference		No Reference			GTTCA		
		PSNR-Y↑	SSIM-Y↑	LPIPS↓	DISTS↓	MANIQA↑	MUSIQ↑	CLIP-IQA↑	Overall↑	Level 1↑	Level 2↑
HR (oracle)		80.0000	1.0000	0.0000	0.0000	0.4261	66.2367	0.5403	94.6813	96.2942	90.6911
A	Real-ESRGAN	24.0453	0.7694	0.2134	0.1719	0.4463	68.5397	0.4611	40.3167	52.9647	9.0268
	SwinIR	24.7762	**0.7859**	**0.1989**	0.1626	0.4288	66.9483	0.4357	45.6760	59.4071	11.7066
B	DiffBIR	24.1097	0.7039	0.2591	0.1880	**0.6127**	**71.8427**	0.6417	54.9736	69.6123	18.7588
	SUPIR	24.2971	0.7098	0.2622	0.1947	0.4608	63.5998	0.5922	55.2984	68.5291	22.5670
	FaithDiff	24.2038	0.7363	0.2061	**0.1623**	0.5095	71.6540	0.5821	52.0097	65.1653	19.4640
	DiT4SR	22.6539	0.6933	0.2477	0.1830	0.4973	70.8502	0.6074	59.1555	73.1471	**24.5416**
C	ResShift	24.9170	0.7645	0.2219	0.1839	0.4110	63.3033	0.4902	47.7060	61.0034	14.8096
	SinSR	**25.1644**	0.7668	0.2116	0.1747	0.4551	66.1994	0.5574	43.1181	55.9863	11.2835
	OSEDiff	23.9596	0.7533	0.2147	0.1666	0.4803	71.1559	0.5672	44.6204	58.4379	10.4372
D	PURE	22.2970	0.6706	0.2652	0.1893	0.5233	70.0036	**0.6531**	34.3890	45.7241	6.3470
E	TeReDiff	23.5188	0.7233	0.2362	0.1796	0.5638	70.5181	0.6183	51.2383	65.1083	16.9252
	UniT	22.7672	0.6503	0.2814	0.2072	0.5159	68.7382	0.6315	**61.3886**	**76.4538**	24.1185
Pearson r vs. Overall		−0.1566	−0.4414	+0.4333	+0.4459	+0.3175	+0.0806	+0.3528	–		
Spearman ρ vs. Overall		−0.1329	−0.5455	+0.4196	+0.4056	+0.3147	+0.1958	+0.3636	–		

contaminate the measurement of text restoration quality, since an instance missed by the detector counts as a failure even if its text is well restored. Building on the reliable annotations of ReasonText, we propose *Ground-Truth Text-Crop Accuracy* (GTTCA), which extends the recognition-based evaluation of STISR [4, 32] to the full-image setting of TAISR. GTTCA crops each text instance with its ground-truth (GT) polygon, removing the detection stage from the measurement.

GTTCA is computed in four steps (see Figure A in Appendix). For each text instance, we (1) crop the SR output around its GT polygon with padding, (2) rotate and flip the crop according to its GT orientation so that the text reads upright, (3) recognize the text in the crop with an OCR model, and (4) check whether the recognized text exactly matches the GT transcript after normalization, following the standard STISR protocol [4, 32]. GTTCA is the fraction of matched instances. We report GTTCA separately on Level 1 and Level 2 instances, as well as over both levels (Overall), using PaddleOCR-VL-1.5 [6] as the OCR model. Further details are provided in Appendix A.2.

3.3 SR Models Fail on Reasoning-Required Text

We evaluate twelve recent publicly available real-world SR and TAISR models on ReasonText, covering five groups: (A) GAN-based [16, 33], (B) multi-step diffusion-based [5, 9, 17, 41], (C) few-step diffusion-based [34, 37, 43], (D) AR-based [35], and (E) TAISR[1] [15, 23]. We report fidelity (PSNR and SSIM on the Y channel), perceptual quality (reference-based LPIPS [45] and DISTS [7], and no-reference MANIQA [39], MUSIQ [14], and CLIP-IQA [30]), and text restoration quality (GTTCA). We also report all metrics on the HR images as a reference (HR oracle). Table 1 reports the results, and qualitative results are shown in Appendix A.3.

Finding 1: All models struggle on reasoning-required text. GTTCA drops sharply from Level 1 to Level 2 for all twelve models, regardless of architecture, training paradigm, or whether the model is explicitly text-aware. The drop is large enough that the score ranges of the two levels do not overlap. The lowest Level 1 score (45.72%) still exceeds the highest Level 2 score (24.54%). Since the HR oracle reaches 90.69% on Level 2, far above every SR model, the gap reflects a limitation of the SR models rather than of the text recognizer behind the metric.

Finding 2: Current text-aware design helps only on locally recoverable text. The TAISR model UniT [15] achieves the best overall GTTCA (61.39%), ahead of the strongest real-world SR model, DiT4SR [9] (59.16%). Since the two build on the same backbone [10], this comparison largely isolates the effect of UniT's text-aware design. The margin, however, comes entirely from Level 1, where UniT leads by over three points (76.45% versus 73.15%), while on Level 2 it vanishes (24.12%

[1] Some prior TAISR models such as TextSR [40] and TADiSR [13] are not publicly released and are therefore excluded from our comparison.

Table 2: **Injecting ground-truth text improves GTTCA for all six text-conditioned SR models.** Each model is evaluated with its original caption and with the GT transcripts of its Level 1 and Level 2 instances inserted as a natural-language caption. In the + GT rows, blue/red indicate an improvement/decline over the original caption. ↑(↓): higher (lower) is better.

Group	Model	Fidelity		Perceptual					Text Restoration		
				Reference		No Reference			GTTCA		
		PSNR-Y↑	SSIM-Y↑	LPIPS↓	DISTS↓	MANIQA↑	MUSIQ↑	CLIP-IQA↑	Overall↑	Level 1↑	Level 2↑
HR (oracle)		80.0000	1.0000	0.0000	0.0000	0.4261	66.2367	0.5403	94.6813	96.2942	90.6911
B	SUPIR	24.2971	0.7098	0.2622	0.1947	0.4608	63.5998	0.5922	55.2984	68.5291	22.5670
	+ GT	24.5912	0.7038	0.2732	0.1964	0.4349	60.9064	0.5714	69.7686	81.6534	40.3667
	FaithDiff	24.2038	0.7363	0.2061	0.1623	0.5095	71.6540	0.5821	52.0097	65.1653	19.4640
	+ GT	24.4677	0.7467	0.1962	0.1567	0.4951	71.4650	0.5739	62.4523	75.3250	30.6065
	DiT4SR	22.6539	0.6933	0.2477	0.1830	0.4973	70.8502	0.6074	59.1555	73.1471	24.5416
	+ GT	23.0546	0.7182	0.2317	0.1797	0.4944	70.8355	0.5932	74.1048	85.1767	46.7137
C	OSEDiff	23.9596	0.7533	0.2147	0.1666	0.4803	71.1559	0.5672	44.6204	58.4379	10.4372
	+ GT	23.9394	0.7546	0.2179	0.1695	0.4811	71.1110	0.5676	51.4657	66.4082	14.4993
E	TeReDiff	23.5188	0.7233	0.2362	0.1796	0.5638	70.5181	0.6183	51.2383	65.1083	16.9252
	+ GT	23.1785	0.7176	0.2388	0.1812	0.5694	71.0449	0.6251	59.5128	72.7708	26.7137
	UniT	22.7672	0.6503	0.2814	0.2072	0.5159	68.7382	0.6315	61.3886	76.4538	24.1185
	+ GT	23.1021	0.6725	0.2694	0.1994	0.5024	68.3612	0.6169	70.6618	82.8848	40.4231

versus 24.54%). This matches our earlier argument that text-aware modules remain centered on local visual evidence, so their benefit does not extend to reasoning-required text.

Finding 3: Better conventional SR quality does not mean better text restoration. We compute the Pearson and Spearman rank correlation between each conventional metric and overall GTTCA across the twelve models. None of them tracks text restoration quality. PSNR and SSIM show no positive correlation with GTTCA. LPIPS and DISTS, for which lower is better, correlate in the wrong direction. No-reference image quality assessment (NR-IQA) metrics correlate only weakly. Improving conventional SR quality therefore does not carry over to text restoration, which has to be measured on its own.

Together, these findings answer the question we posed: *TAISR does need reasoning.* What remains is how to supply it, which we address next.

4 Reasoning Transfer via Captioning

Reasoning can happen inside the SR model or outside it. The internal approach gives the SR model itself the ability to reason. The external approach obtains reasoning from an external model and injects it into the SR model. The internal approach is appealing but currently hard to pursue. Finding 3 shows that the conventional quality criteria that SR training pursues do not track text restoration, and Finding 2 shows that even text-specific modules and objectives do not help on reasoning-required text. This leaves no obvious training signal for teaching an SR model to infer locally illegible text, which motivates our external approach.

This approach raises two questions.

(i) **Reasoning generation.** What kind of reasoning-aware representation should we extract from the LR image?

(ii) **Injection.** How should this representation be injected into the SR model so that it restores reasoning-required text?

The design space for both questions is broad. A reasoning-aware representation could take many forms, such as a similar image retrieved from an external source, a text string inferred from the LR image, or features taken from the reasoning model itself. Each of these could in turn be injected in different ways, for example by concatenating it with the SR model's intermediate features, attending to it in a cross-attention layer, or passing it as the model's text input. Rather than exploring this space exhaustively, we derive a method from two empirical observations. The first settles both design questions, and the second makes the resulting choice practical.

Table 3: **The reasoning prompt substantially improves MLLMs on reasoning-required (Level 2) text, where OCR models and the no-reasoning prompt lag far behind.** (Left) Text recall of dedicated OCR models. (Right) Text recall of MLLMs under the no-reasoning / reasoning prompts. Best per column in **bold**. ↑: higher is better.

Model	Overall↑	Level 1↑	Level 2↑
PaddleOCR-VL-1.5	48.03	54.90	31.03
TESTR	56.44	61.29	**44.43**
GLM-OCR	**59.32**	**66.08**	42.60

Model	no-reasoning			reasoning		
	Overall↑	Level 1↑	Level 2↑	Overall↑	Level 1↑	Level 2↑
Qwen2.5-VL 7B	46.98	56.84	22.57	58.51	66.08	39.77
Qwen3.5 27B	58.42	69.67	30.61	77.51	83.81	61.92
Gemma 4 31B	**69.18**	**78.62**	**45.84**	**82.01**	**86.72**	**70.38**

4.1 Observation 1: Ground-Truth Transcripts Improve Text Restoration

Many recent SR models [5, 9, 15, 23, 37, 41] already accept a text input, originally intended to convey high-level semantics or, in TAISR models, predicted text. Since this channel is already in place, we examine whether it can carry reasoning-aware information.

To test this, we inject the GT transcripts of ReasonText directly into the caption. Given the n transcripts $t_1, t_2, \ldots, t_n$ of the Level 1 and Level 2 instances in an image, we replace the default caption with a natural-language caption listing them, of the form `"The image with the text ‘`t_1`’, ‘`t_2`’, …, ‘`t_n`’."`, and re-evaluate on ReasonText. Table 2 shows that GTTCA improves for every model on both levels. The gains on Level 2 are substantial. DiT4SR [9] gains over 22 points, from 24.54% to 46.71%, and SUPIR [41] and UniT [15] gain over 16 points each. As a side effect, NR-IQA scores drop for several models. We attribute the drop to the caption no longer describing the scene, since these models are trained with scene-describing captions. Adding a scene description largely restores NR-IQA (Appendix B.1), and the RTC caption therefore includes one alongside the inferred text (Section 4.3).

This observation settles both design questions. For injection, we pass the reasoning-aware representation through the text channel that text-conditioned SR models already provide. For reasoning generation, the reasoning-aware representation is a natural-language caption listing the GT transcripts of the image. In a real-world scenario, however, the GT transcripts of an LR image are unavailable, so we need a way to approximate this caption from the LR image alone, which we address next.

4.2 Observation 2: MLLMs Can Approximate Ground-Truth Transcripts

Choosing a reasoning generator. How can we obtain GT transcripts from an LR image? We measure a model's ability to do so with text recall, the fraction of GT transcripts that exactly match a word in its output, after both are normalized. We start with OCR models [6, 8, 46], the standard tool for reading text from images, and run them on the LR images of ReasonText. Table 3 shows that their recall drops sharply on Level 2, where even the best model reaches only 44.43%. Despite the rich priors from large image-text corpora, OCR models struggle with reasoning-required text, which makes them unsuitable as a reasoning generator.

We therefore turn to MLLMs, which read text as OCR models do but can also reason about the image [21, 42]. We run Qwen2.5-VL [2], which UniT adopts, together with two more recent MLLMs, Qwen3.5 [29] and Gemma 4 [12]. Each model is run under two prompts. The *no-reasoning prompt* instructs the model to list the text it can see, and the *reasoning prompt* additionally instructs it to infer text using image context, logical patterns, or prior knowledge.

Table 3 shows that the reasoning prompt improves every model at both levels, and the gain is larger on Level 2 than on Level 1 in all three cases. Gemma 4 reaches the highest score, 70.38% on Level 2, which exceeds the best OCR model by 25.95 points. It misses some reasoning-required text, but reads a large portion of it. We therefore adopt Gemma 4 with the reasoning prompt as our reasoning generator, and use the transcripts it reads to build the caption.

Does the gain come from reasoning? One might ask whether the high Level 2 recall of Gemma 4 comes from something other than reasoning, and two explanations stand out. It may be exceptional local text recognition, strong enough to read from the text region alone what our annotators could not. Or it may be memorization, since ReasonText is built on the publicly available RealSR-v3 and DRealSR, which the model could have seen during pre-training.

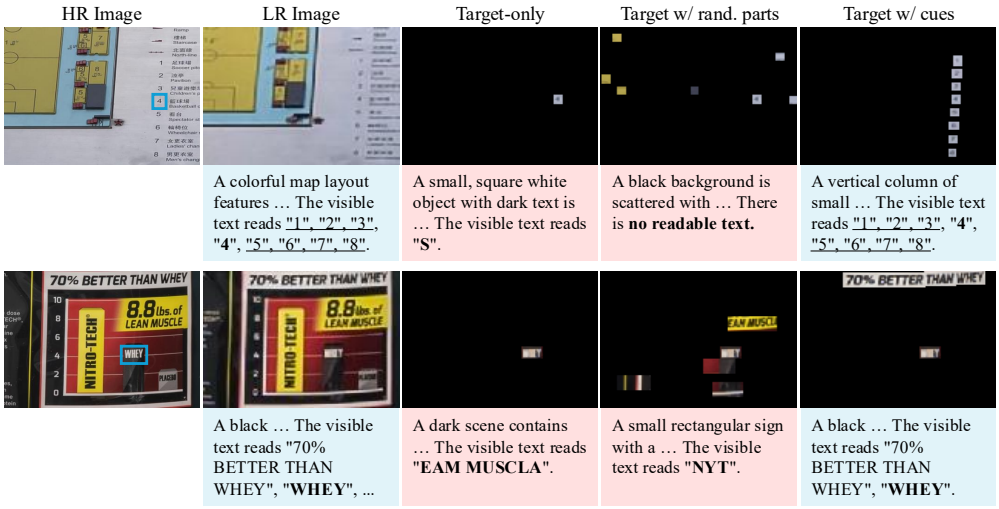


Figure 3: **Gemma 4 reads locally illegible text by reasoning over relevant cues.** Top: given only the target region (the locally illegible "4"), Gemma 4 fails. Adding random unrelated patches does not help, but adding the related numeric column "1"–"8" restores the correct reading. Bottom: likewise, "WHEY" is read only when the banner "70% BETTER THAN WHEY" is visible. Blue and red backgrounds mark correct and incorrect readings.

To test the first, we mask everything outside the GT text regions, which removes the surrounding scene while leaving the text pixels intact, and run the same model on the masked images. The effect differs sharply between the two levels. Level 1 recall drops only slightly, from 86.72% to 82.55%, since locally recoverable text does not need the surroundings. In contrast, Level 2 recall collapses, from 70.38% to 29.90%. If Gemma 4 were simply a strong local text recognizer, masking the surroundings would barely change Level 2 recall, just as it barely changes Level 1 recall. The collapse rules out the first explanation, since Gemma 4 clearly needs the surroundings to read reasoning-required text.

To test the second, we vary the visible surrounding region (Figure 3). Gemma 4 recovers the correct transcript on **Target w/ cues**, where only the target text and the relevant cues are visible. In contrast, it fails on **Target w/ rand. parts**, where the cues are replaced by random regions of the same area, and on **Target-only**, the masked setting used above. This argues against the second explanation, since a model recalling a memorized image would not depend on the cue region in particular. With both alternatives unlikely, reasoning over the surroundings remains the most plausible explanation.

4.3 Method: Reasoning Transfer via Captioning

Section 4.1 showed that a caption carrying the GT transcripts improves the restoration of reasoning-required text, and Section 4.2 showed that an MLLM with the reasoning prompt can infer most of those transcripts from the LR image. Putting the two together gives *Reasoning Transfer via Captioning* (RTC). Given an LR image $\mathbf{x}_{\text{LR}}$, RTC proceeds in two steps.

1. **Caption by reasoning.** Query an MLLM with $\mathbf{x}_{\text{LR}}$ and the reasoning prompt, which returns a two sentence caption c: a scene description followed by the inferred transcripts.

2. **Restore by conditioning.** Pass $(\mathbf{x}_{\text{LR}}, c)$ to a text-conditioned SR model, with c replacing its default caption, to produce the restored image $\hat{\mathbf{x}}_{\text{HR}}$.

Since both models remain frozen and the caption c is the only interface between them, RTC is training-free and attaches to any text-conditioned SR model, whether it was designed for real-world SR or for TAISR. This interface also makes failures traceable, since we can check whether the MLLM misread the text or the SR model failed to render it.

Table 4: **Quantitative comparison of text-conditioned SR models with and without RTC on ReasonText.** RTC consistently improves text restoration for all six models. Baselines use each model's default caption source. In the + RTC rows, blue/red indicate an improvement/decline over the baseline. ↑(↓): higher (lower) is better.

Group	Model	Fidelity		Perceptual					Text Restoration		
				Reference		No Reference			GTTCA		
		PSNR-Y↑	SSIM-Y↑	LPIPS↓	DISTS↓	MANIQA↑	MUSIQ↑	CLIP-IQA↑	Overall↑	Level 1↑	Level 2↑
HR (oracle)		80.0000	1.0000	0.0000	0.0000	0.4261	66.2367	0.5403	94.6813	96.2942	90.6911
B	SUPIR	24.2971	0.7098	0.2622	0.1947	0.4608	63.5998	0.5922	55.2984	68.5291	22.5670
	+ RTC	23.9534	0.7071	0.2480	0.1846	0.5202	66.8668	0.6412	67.3163	78.8483	38.7870
	FaithDiff	24.2038	0.7363	0.2061	0.1623	0.5095	71.6540	0.5821	52.0097	65.1653	19.4640
	+ RTC	24.2724	0.7387	0.2020	0.1602	0.4975	71.4577	0.5797	61.0231	74.0023	28.9140
	DiT4SR	22.6539	0.6933	0.2477	0.1830	0.4973	70.8502	0.6074	59.1555	73.1471	24.5416
	+ RTC	22.6816	0.6936	0.2479	0.1830	0.5009	70.6745	0.6110	73.5688	84.7777	45.8392
C	OSEDiff	23.9596	0.7533	0.2147	0.1666	0.4803	71.1559	0.5672	44.6204	58.4379	10.4372
	+ RTC	23.8903	0.7504	0.2171	0.1682	0.4886	71.2281	0.5767	49.4113	64.2531	12.6939
E	TeReDiff	23.5188	0.7233	0.2362	0.1796	0.5638	70.5181	0.6183	51.2383	65.1083	16.9252
	+ RTC	23.3884	0.7212	0.2377	0.1795	0.5777	70.9852	0.6237	56.6382	70.1254	23.2722
	UniT	22.7672	0.6503	0.2814	0.2072	0.5159	68.7382	0.6315	61.3886	76.4538	24.1185
	+ RTC	22.8425	0.6555	0.2751	0.2041	0.5169	68.8130	0.6338	70.9704	82.5542	42.3131

5 Experiments

5.1 Experimental Settings

We attach RTC to the six text-conditioned SR models of Section 4.1. For each model, the baseline is its official inference pipeline with its default caption source, such as LLaVA-1.5 for DiT4SR and Qwen2.5-VL for UniT. *+RTC* replaces only this caption with the one Gemma 4 produces under the reasoning prompt, leaving everything else unchanged. For a controlled comparison, we precompute one caption per LR image and feed the same caption to all six models, keeping it fixed throughout denoising. We evaluate on ReasonText with the metrics of Section 3.3 and run all experiments on NVIDIA H200 GPUs. Table 4 reports the results, and Figure 4 shows qualitative comparisons.

5.2 Results

RTC recovers most of the gains observed with GT transcripts. RTC replaces the GT transcripts with transcripts inferred by an MLLM, so we first check whether the observations made with GT transcripts still hold under this replacement. Table 4 shows that they do. RTC improves GTTCA for every model on both levels, whether the model was designed for real-world SR or for TAISR. As with GT transcripts, the gains are larger on Level 2 than on Level 1 for most models, reaching 21.30 points for DiT4SR (from 24.54% to 45.84%). Moreover, the scene description in the RTC caption largely avoids the NR-IQA drop observed with transcript-only captions, and fidelity and reference-based perceptual metrics also remain stable. RTC is thus a practical method that needs neither GT transcripts nor training, and it improves the restoration of reasoning-required text, where all SR models struggle, while preserving general image quality.

With RTC, a real-world SR model surpasses a TAISR model. Without RTC, UniT achieves the best overall GTTCA, ahead of DiT4SR, which shares the same backbone but has no text-aware module. With RTC, however, the order reverses. DiT4SR + RTC outperforms UniT + RTC on both levels, even though both receive the same caption. This suggests that the current text-aware design is not complementary to reasoning, since a model without it performs better once reasoning is supplied. We therefore argue for rethinking text-aware design with reasoning as its core axis.

Generalization and robustness of RTC. The gains of RTC are not specific to ReasonText. On the existing TAISR benchmarks, SA-Text-test and RealText, RTC improves text restoration for all six models under their original evaluation protocol. RTC is also robust to imperfect captions. Even when Gemma 4 misreads the text with spelling errors, RTC still improves GTTCA, since the SR models can often restore the correct text. RTC therefore does not lower text restoration accuracy even when the MLLM misreads the text. Detailed results are provided in Appendix C.

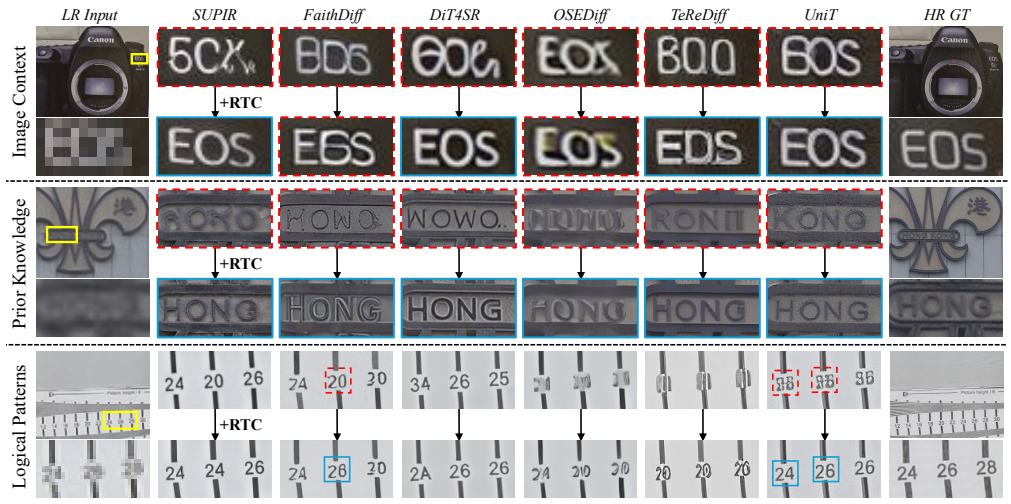


Figure 4: **Qualitative results of RTC across reasoning types.** For Level 2 instances of each reasoning type, the top and bottom rows show the outputs of six text-conditioned SR models without and with RTC, respectively. With RTC, some models can turn incorrect text into correct text.

6 Conclusion and Future Work

We asked whether text-aware image super-resolution needs reasoning. To answer this, we introduced ReasonText, a benchmark that separates reasoning-required text from locally recoverable text, and GTTCA, a metric that measures text restoration quality without detection errors. With these tools, we found that twelve recent SR models, including those designed for text, all struggle on reasoning-required text. As a remedy, we proposed RTC, a training-free plug-in, based on two observations: (i) text-conditioned SR models restore text substantially better when their captions contain the GT transcripts, and (ii) MLLMs can infer these transcripts from the LR image through reasoning. In our experiments, RTC consistently improves text restoration for both real-world SR and TAISR models, especially on reasoning-required text. Moreover, with RTC, a real-world SR model surpasses a TAISR model built on the same backbone, which suggests that reasoning should be a core axis of text-aware design.

While our study shows that TAISR needs reasoning, several limitations remain. First, ReasonText is human-curated and built on existing real-world SR benchmarks, so it is limited in size and scope. It contains 513 LR–HR pairs of natural images and annotates only alphanumeric text, leaving other scripts unannotated. Future work can extend it to more languages, a larger scale, more diverse domains, and revisit Level 3 with a larger or more specialized annotator pool. Second, even the best model with RTC remains far below the HR oracle on Level 2 (45.84% versus 90.69%). Most of this gap therefore lies not in reading the text but in rendering it, calling for SR models that faithfully render the given text as well as better ways of bringing reasoning into SR, including internal approaches. Finally, RTC applies only to text-conditioned SR models and cannot be used with SR models that have no text channel. A more general way of injecting reasoning, applicable to any SR model, is left for future work.

Acknowledgement

This work was supported by Samsung Electronics Co., Ltd. (IO251211-14355-01), Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT2501-02, and Artificial Intelligence Graduate School Program, Yonsei University (No. RS-2020-II201361).

References

[1] 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.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3086–3095, 2019.

[4] Jingye Chen, Bin Li, and Xiangyang Xue. Scene text telescope: Text-focused scene image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2021.

[5] Junyang Chen, Jinshan Pan, and Jiangxin Dong. Faithdiff: Unleashing diffusion priors for faithful image super-resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28188–28197, 2025.

[6] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr-vl-1.5: Towards a multi-task 0.9b vlm for robust in-the-wild document parsing, 2026. URL `https://arxiv.org/abs/2601.21957`.

[7] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

[8] Shuaiqi Duan, Yadong Xue, Weihan Wang, Zhe Su, Huan Liu, Sheng Yang, Guobing Gan, Guo Wang, Zihan Wang, Shengdong Yan, Dexin Jin, Yuxuan Zhang, Guohong Wen, Yanfeng Wang, Yutao Zhang, Xiaohan Zhang, Wenyi Hong, Yukuo Cen, Da Yin, Bin Chen, Wenmeng Yu, Xiaotao Gu, and Jie Tang. Glm-ocr technical report, 2026. URL `https://arxiv.org/abs/2603.10910`.

[9] Zheng-Peng Duan, Jiawei Zhang, Xin Jin, Ziheng Zhang, Zheng Xiong, Dongqing Zou, Jimmy S Ren, Chunle Guo, and Chongyi Li. Dit4sr: Taming diffusion transformer for real-world image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18948–18958, 2025.

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Google DeepMind. Gemma 4 31b. `https://huggingface.co/google/gemma-4-31B`, 2026. Hugging Face model card. Accessed: 2026-05-05.

[13] Qiming Hu, Linlong Fan, Yiyan Luo, Yuhang Yu, Xiaojie Guo, and Qingnan Fan. Text-aware real-world image super-resolution via diffusion model with joint segmentation decoders. *arXiv preprint arXiv:2506.04641*, 2025.

[14] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[15] Jin Hyeon Kim, Paul Hyunbin Cho, Claire Kim, Jaewon Min, Jaeeun Lee, Jihye Park, Yeji Choi, and Seungryong Kim. Unified diffusion transformer for high-fidelity text-aware image restoration. *arXiv preprint arXiv:2512.08922*, 2025.

[16] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

[17] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European conference on computer vision*, pages 430–448. Springer, 2024.

[18] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yi Xin, Xinyue Li, Qi Qin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

[19] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[20] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8048–8064, 2021.

[21] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[22] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019.

[23] Jaewon Min, Jin Hyeon Kim, Paul Hyunbin Cho, Jaeeun Lee, Jihye Park, Park Min Kyu, Sangpil Kim, Hyunhee Park, and Seungryong Kim. Text-aware image restoration with diffusion models. In *The Fourteenth International Conference on Learning Representations*.

[24] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[25] Junbo Qiao, Miaomiao Cai, Wei Li, Xudong Huang, Jie Hu, Xinghao Chen, Shaohui Lin, and Hongkai Xiong. Realsr-r1: Reinforcement learning for real-world image super-resolution with vision-language chain-of-thought. *arXiv preprint arXiv:2506.16796*, 2025.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[27] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.

[28] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

[29] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

[30] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2555–2563, 2023.

[31] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024.

[32] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. Scene text image super-resolution in the wild. In *European conference on computer vision*, pages 650–666. Springer, 2020.

[33] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021.

[34] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25796–25805, 2024.

[35] Hongyang Wei, Shuaizheng Liu, Chun Yuan, and Lei Zhang. Perceive, understand and restore: Real-world image super-resolution with autoregressive multimodal generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18640–18650, 2025.

[36] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *European conference on computer vision*, pages 101–117. Springer, 2020.

[37] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37: 92529–92553, 2024.

[38] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024.

[39] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1191–1200, 2022.

[40] Keren Ye, Ignacio Garcia Dorado, Michalis Raptis, Mauricio Delbracio, Irene Zhu, Peyman Milanfar, and Hossein Talebi. Textsr: diffusion super-resolution with multilingual ocr guidance. *arXiv preprint arXiv:2505.23119*, 2025.

[41] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25669–25680, 2024.

[42] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[43] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in neural information processing systems*, 36:13294–13307, 2023.

[44] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824. IEEE, 2023.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[46] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9519–9528, 2022.

A Additional Details on TAISR Needs Reasoning

A.1 ReasonText Benchmark

This section details the construction of ReasonText (Section 3.1).

HR image annotation. Three annotators participate in HR image annotation, and both annotation tasks below follow the same procedure. One annotator labels all instances in a pass, and a second annotator reviews the pass and corrects any label they find incorrect. The two then alternate, each reviewing the other's corrections, until a pass ends with no change. A third annotator reviews the converged labels, and any instance they flag is resolved by discussion among all three.

1. **Text polygons.** Annotators draw a word-level polygon around every legible alphanumeric text instance in each HR image, including curved and perspective-distorted ones. Each polygon uses as many points as needed to enclose the text and nothing else.
2. **Transcripts and orientations.** Annotators transcribe each text instance and record the transformation that brings it upright, consisting of a rotation from $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ and an optional horizontal flip.

LR–HR patch construction. Existing TAISR benchmarks [23] obtain LR–HR pairs by random cropping. We find that random cropping often removes the surrounding context that humans use to read locally unreadable text, turning a reasoning-required instance into an unrecoverable one. We therefore crop patches manually, with three annotators participating in this stage.

For each text instance in an HR image, annotators draw a square crop that includes any content related to that instance, and record its coordinates. When no such content is present, the annotators draw a random square crop containing the instance. Two constraints apply: the crop must be at least 512×512, and it must enclose the target text polygon.

Since a crop is drawn for every text instance, many of the resulting crops overlap. Among overlapping crops with similar size, we automatically keep the one containing the most text instances and discard the others. We then map the coordinates of each retained crop to the $\times 4$ LR image provided by RealSR-v3 [3] and DRealSR [36] by dividing them by four, and crop the corresponding LR region. Since crop sizes vary, we finally resize the HR crop to 512×512 and the LR crop to 128×128, obtaining the standard $\times 4$ SR setting used in prior work [15, 23].

HR patch re-annotation. Cropping can cut a text instance at the patch boundary, leaving a polygon that extends beyond the patch or a word that is only partially visible. We therefore re-annotate the instances that cross a patch boundary, adjusting each polygon to the visible extent and relabeling its transcript accordingly. This re-annotation follows the same procedure as the HR image annotation, with two annotators alternating corrections until a pass ends with no change, followed by a review by a third annotator whose flagged instances are resolved by discussion among all three.

Design choices for difficulty labeling. We make three core design choices that apply to both stages of Section 3.1. (1) We assign a level to an instance when any of the five annotators identifies it, rather than requiring agreement among them. (2) We select ten annotators for whom English is either a native language or a primary second language. Otherwise, an annotator unfamiliar with English would fail to read instances that others can read, pushing these instances to a higher level and shrinking the set of instances the benchmark can evaluate. (3) We require all labeling sessions to be conducted offline without internet access. Since ReasonText is built on the publicly available RealSR-v3 [3] and DRealSR [36], searching the web with an LR patch could retrieve its HR counterpart and expose the transcript. An instance resolved this way would be recorded as readable even when no cue in the image supports it, mixing looked-up text into the reasoning-required set.

Cue aggregation in Stage 2. A reasoning-required instance can be read through more than one route. For example, one annotator may identify a text from a repeated word elsewhere in the patch, while another identifies the same text from a brand they recognize. We therefore do not assign a single cue to an instance. Instead, we take the union of the cues selected by the annotators who identified the instance.

Table A: **Reliability of the ReasonText difficulty levels.** (a) Agreement, the fraction of annotators who attempted an instance and read it correctly, for the Level 1 and Level 2 instances assigned under our rule. The rows give the share of instances in each agreement range, followed by the mean and median. (b) Level counts re-derived by requiring at least one, at least half, more than half, or all of the attempting annotators to read an instance correctly. Agreement is high at both levels, and Level 2 does not shrink under stricter thresholds but grows from 709 to 885 under unanimity.

(a) Inter-annotator agreement

Agreement	L1 (N=1754)	L2 (N=709)
all correct	71.9%	73.6%
50–99%	24.1%	19.5%
<50%	3.9%	6.9%
mean	89.3%	87.5%
median	100%	100%

(b) Level counts by threshold

Threshold	L1	L2	L3
≥1 correct (ours)	1754	709	1522
majority (≥50%)	1685	717	1583
strict (>50%)	1571	737	1677
unanimous (100%)	1262	885	1838

Level 3 text instances. Level 3 is defined by what our annotators could not read, not by what is inherently unreadable. A larger annotator pool or annotators with specific domain knowledge might recover text that no one in our pool could. We therefore release Level 3 instances with their polygons and transcripts, so that future work can revisit their levels with different annotator populations.

Statistics. ReasonText contains 513 LR–HR pairs with 3,985 annotated text instances, of which the 2,463 Level 1 and Level 2 instances are used for main experiments. The instances are distributed as follows:

Level	# instances	%
Level 1 (locally recoverable)	1,754	44.0
Level 2 (reasoning-required)	709	17.8
Level 3 (unrecoverable)	1,522	38.2
Total	3,985	100.0

Among the 709 Level 2 instances, 414 are labeled with prior knowledge, 288 with image context, and 246 with logical patterns. These counts sum to more than 709 because an instance carries the union of the cues selected by the annotators who identified it.

Robustness of the level assignment. We assign a level when any of the five annotators identifies an instance. Each instance has a GT transcript from its HR image, so what a level should capture is whether the text is recoverable at all, not how many annotators happen to recover it. Annotators differ in attention and in the prior knowledge they bring, and one correct reading is enough to establish that a route to the text exists. A majority rule would instead push text recovered by only some annotators to a harder level, making it appear harder to recover than it is.

Table A(a) reports agreement, defined for each instance as the fraction of attempting annotators whose transcript matches the GT transcript, for the levels assigned under our rule. We exclude those who skipped, since they are instructed to skip whenever they are unsure. Agreement averages 89.3% for Level 1 and 87.5% for Level 2, with a median of 100% for both, and all attempting annotators read the text correctly for 71.9% of Level 1 and 73.6% of Level 2 instances. Agreement is thus high at both levels, and the assigned levels are stable.

Table A(b) re-derives the levels by re-running the two-stage assignment under stricter thresholds, which count an instance as identified only when at least half, more than half, or all of its attempting annotators read it correctly. Under the majority rule, the share of each level changes by less than two percentage points. Even under unanimity, Level 2 does not shrink but grows from 709 to 885, as instances that leave Level 1 are re-examined in Stage 2 and recovered there with the surrounding context. The reasoning-required set therefore persists under every threshold we tested.

Re-evaluating the SR models on the levels re-derived under the majority rule leads to the same conclusions (Table B). SR models still fail on reasoning-required text: GTTCA drops sharply from

Table B: **The conclusions hold under the majority rule.** GTTCA on ReasonText with the levels re-derived under the majority ($\geq$50%) rule of Table A (b), which gives 1,685 Level 1 and 717 Level 2 instances. All twelve models drop sharply on Level 2, RTC improves all six text-conditioned SR models, and DiT4SR + RTC outperforms UniT + RTC. Blue marks an improvement over the baseline.

Group	Model	Baseline			+ RTC		
		Overall↑	Level 1↑	Level 2↑	Overall↑	Level 1↑	Level 2↑
A	Real-ESRGAN	41.0	53.9	10.7	–	–	–
	SwinIR	46.4	60.1	14.2	–	–	–
B	DiffBIR	55.7	70.9	20.2	–	–	–
	SUPIR	55.8	69.3	24.1	67.8	79.5	40.2
	FaithDiff	52.5	65.9	20.8	61.6	74.8	30.7
	DiT4SR	60.0	74.1	26.9	74.0	85.0	48.1
C	ResShift	48.4	61.8	17.0	–	–	–
	SinSR	43.8	56.9	13.1	–	–	–
	OSEDiff	45.2	59.1	12.7	49.9	65.0	14.5
D	PURE	35.0	46.4	8.2	–	–	–
E	TeReDiff	52.0	65.9	19.2	57.2	71.0	24.8
	UniT	61.8	76.9	26.2	71.2	83.1	43.2

Level 1 to Level 2 for every model, and the lowest Level 1 score (46.4%) still far exceeds the highest Level 2 score (26.9%). RTC is effective as well, improving all six text-conditioned SR models at both levels, with larger gains on Level 2 for most models. Moreover, the real-world SR model DiT4SR with RTC again outperforms the TAISR model UniT with RTC.

A.2 GTTCA Metric

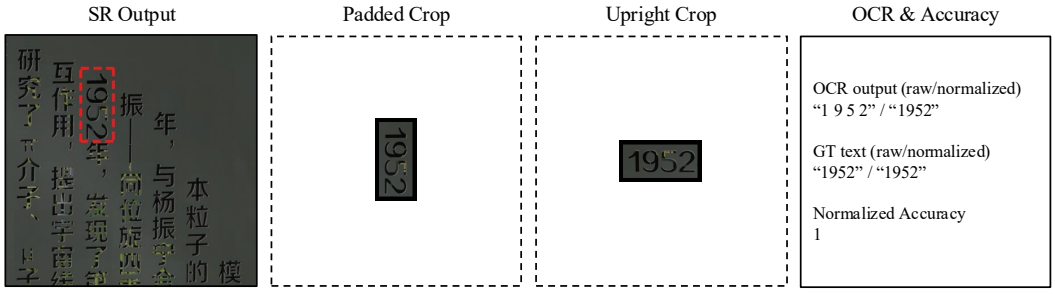


Figure A: **Computing GTTCA on a restored text instance.** Each text instance is cropped around its GT polygon with padding, rotated and flipped with its GT orientation, and recognized by an OCR model. The recognized text is compared with the GT transcript under normalized exact match.

Figure A illustrates the four steps of GTTCA for a single text instance. We further validate the design of GTTCA (Section 3.2) with two experiments.

Ablation of the GTTCA protocol. GTTCA differs from an end-to-end text spotting protocol in two ways: it uses GT polygons instead of a text detector, and it rotates and flips each crop with the GT orientation so that the text reads upright. Table C removes these two components one after another, with all protocols using PaddleOCR-VL-1.5 [6] so that any change comes from the protocol rather than the recognizer. The **w/o RF** (without rotation and flipping) protocol keeps the GT polygons but recognizes each crop in its original orientation. The **OCR** protocol additionally drops the GT polygons and use PaddleOCR-VL-1.5 in its end-to-end mode, which detects the text and then recognizes it.

Both components remove a factor that has nothing to do with restoration quality. Without GT polygons, the score depends on whether the detector finds the text: the **OCR** protocol reaches only

Table C: **Both GT polygons and GT orientations are needed for GTTCA to reflect restoration alone.** All three protocols share the same OCR backbone [6]. GTTCA uses both GT polygons and GT orientations, w/o RF uses GT polygons without rotation and flipping, and OCR detects and recognizes text end-to-end without either. The HR oracle row shows the ceiling of each protocol on images that need no restoration. Excluding the HR oracle, **bold** marks the best score in each column, and underline marks the second-best.

Group	Model	Overall			Level 1			Level 2		
		GTTCA	w/o RF	OCR	GTTCA	w/o RF	OCR	GTTCA	w/o RF	OCR
HR (oracle)		94.7	92.7	29.0	96.3	94.7	30.0	90.7	87.6	26.7
A	Real-ESRGAN	40.3	38.2	12.0	53.0	50.7	16.6	9.0	7.3	0.6
	SwinIR	45.7	43.3	13.9	59.4	57.0	18.6	11.7	9.3	2.3
B	DiffBIR	55.0	53.5	15.1	69.6	67.8	20.0	18.8	17.9	3.0
	SUPIR	55.3	54.0	16.8	68.5	67.1	21.6	22.6	21.4	4.9
	FaithDiff	52.0	50.4	15.6	65.2	63.4	20.1	19.5	18.2	4.4
	DiT4SR	59.2	57.8	17.0	73.1	71.7	21.3	**24.5**	23.4	**6.2**
C	ResShift	47.7	45.5	14.3	61.0	58.7	19.6	14.8	13.0	1.4
	SinSR	43.1	41.1	12.8	56.0	53.6	17.5	11.3	10.2	1.1
	OSEDiff	44.6	43.4	12.5	58.4	57.2	17.1	10.4	9.0	1.3
D	PURE	34.4	33.2	10.4	45.7	44.1	14.2	6.3	6.1	1.1
E	TeReDiff	51.2	49.8	16.1	65.1	63.6	21.6	16.9	15.8	2.5
	UniT	**61.4**	**59.7**	**17.2**	**76.5**	**74.3**	**22.4**	24.1	**23.7**	4.2

Table D: **GTTCA under different OCR recognizers.** Overall GTTCA recomputed with PaddleOCR-VL-1.5 (**Paddle**), ASTER, and MORAN. The weaker recognizers lower every score by a similar proportion, so the ranking of the models barely changes (Table E). In the + RTC columns, blue marks an improvement over the baseline under the same recognizer.

Group	Model	Baseline (Overall)			+ RTC (Overall)		
		Paddle	ASTER	MORAN	Paddle	ASTER	MORAN
A	Real-ESRGAN	40.3	25.8	26.8	–	–	–
	SwinIR	45.7	30.9	30.8	–	–	–
B	DiffBIR	55.0	38.0	40.1	–	–	–
	SUPIR	55.3	38.3	41.7	67.3	49.2	50.9
	FaithDiff	52.0	36.4	38.3	61.0	43.6	45.0
	DiT4SR	59.2	41.7	44.8	73.6	54.7	56.1
C	ResShift	47.7	33.8	33.2	–	–	–
	SinSR	43.1	31.9	31.5	–	–	–
	OSEDiff	44.6	29.7	32.7	49.4	33.0	35.8
D	PURE	34.4	24.8	25.9	–	–	–
E	TeReDiff	51.2	35.0	37.7	56.6	38.8	41.0
	UniT	61.4	43.7	46.3	71.0	49.7	52.3

29.0% on the HR images, which need no restoration at all, and compresses the twelve models into a seven-point range (10.4% to 17.2%). The effect is most severe on Level 2, where eight of the twelve models score 3.0% or lower, leaving no room to compare them on reasoning-required text. Without rotation and flipping, the score depends on how the text happens to be oriented in the image. The **w/o RF** protocol lowers overall GTTCA by 1.2 to 2.4 points for every model and by 2.0 points for the HR oracle (94.7% to 92.7%), and the loss is uneven enough to change the ranking, even swapping DiT4SR and UniT at the top of Level 2.

Robustness to the OCR model. GTTCA reads restored text with an OCR model, which raises the question of whether the results would hold with a different recognizer. We therefore recompute

Table E: **Ranking agreement across recognizers.** Kendall τ and Spearman ρ between the model ranking under PaddleOCR-VL-1.5 and those under ASTER and MORAN, computed on Overall GTTCA in Table D. 1.0 denotes an identical ranking. Both measures stay near 1.0, so GTTCA ranks the SR models in nearly the same order regardless of which recognizer reads the output.

Ranking (Overall GTTCA)	τ vs ASTER	τ vs MORAN	ρ vs ASTER	ρ vs MORAN
12 SR models	0.94	0.94	0.98	0.98
+ RTC (6 SR models)	1.00	1.00	1.00	1.00

GTTCA with ASTER [27] and MORAN [22], two recognizers standard in STISR evaluation, and report the results in Table D.

Since both recognizers are older and weaker than PaddleOCR-VL-1.5 [6], absolute scores fall as expected. However, the drop is similar in proportion (24% to 36%), so the ranking is nearly preserved, with Kendall $\tau \geq 0.94$ and Spearman $\rho \geq 0.98$ against PaddleOCR-VL-1.5 for both recognizers (Table E). GTTCA is therefore robust to the choice of recognizer, since a weaker recognizer lowers every score but barely changes which models restore text better.

A.3 SR Models Fail on Reasoning-Required Text

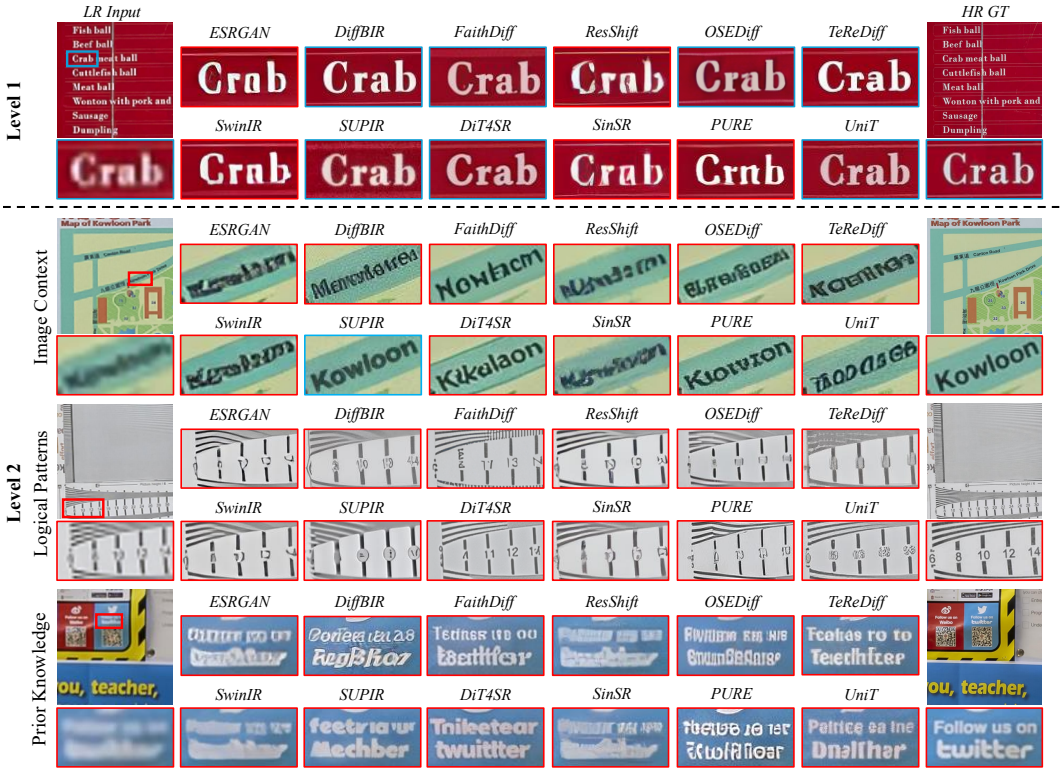


Figure B: **SR models fail on reasoning-required text.** Outputs of the twelve SR models on one Level 1 (locally recoverable) instance and three Level 2 (reasoning-required) instances, one for each cue. Blue and red borders on the SR outputs mark correct and incorrect restorations. On Level 1, most models restore "Crab" correctly. On Level 2, most models fail to restore the text.

Figure B extends Section 3.3 with qualitative comparisons of the twelve SR models on one Level 1 instance and three Level 2 instances, one for each cue. On Level 1, most models restore "Crab" correctly. On Level 2, however, most models fail regardless of architecture or text-awareness, visually echoing the GTTCA drop reported in Table 1.

Table F: **The natural-language caption restores text best among the three caption formats.** Each text-conditioned SR model is evaluated with its original caption and with the GT transcripts of its Level 1 and Level 2 instances injected in three formats. Within each model block, **bold** marks the best value for each metric, and blue/red indicate improvement/decline over the original caption. The HR oracle row reports each metric on the HR images for reference. ↑(↓): higher (lower) is better.

Group	Model	Fidelity		Perceptual					Text Restoration		
				Reference		No Reference			GTTCA		
		PSNR-Y↑	SSIM-Y↑	LPIPS↓	DISTS↓	MANIQA↑	MUSIQ↑	CLIP-IQA↑	Overall↑	Level 1↑	Level 2↑
	HR (oracle)	80.0000	1.0000	0.0000	0.0000	0.4261	66.2367	0.5403	94.6813	96.2942	90.6911
B	SUPIR	24.2971	**0.7098**	**0.2622**	**0.1947**	**0.4608**	**63.5998**	**0.5922**	55.2984	68.5291	22.5670
	+ tag style GT (space)	24.7121	0.7014	0.2840	0.2013	0.4332	60.6653	0.5659	64.9695	77.9590	32.8350
	+ tag style GT (comma)	**24.7246**	0.7006	0.2852	0.2024	0.4291	60.2502	0.5620	64.6935	78.1984	31.2835
	+ caption style GT	24.5912	0.7038	0.2732	0.1964	0.4349	60.9064	0.5714	**69.7686**	**81.6534**	**40.3667**
	FaithDiff	24.2038	0.7363	0.2061	0.1623	**0.5095**	**71.6540**	**0.5821**	52.0097	65.1653	19.4640
	+ tag style GT (space)	24.3528	0.7407	0.2018	0.1595	0.4871	71.2038	0.5758	61.5510	74.6522	29.1396
	+ tag style GT (comma)	24.3674	0.7415	0.2021	0.1600	0.4926	71.3235	0.5777	61.2667	74.7320	27.9549
	+ caption style GT	**24.4677**	**0.7467**	**0.1962**	**0.1567**	0.4951	71.4650	0.5739	**62.4523**	**75.3250**	**30.6065**
	DiT4SR	22.6539	0.6933	0.2477	0.1830	0.4973	70.8502	**0.6074**	59.1555	73.1471	24.5416
	+ tag style GT (space)	22.8881	0.7122	0.2360	0.1814	**0.5015**	**71.1707**	0.5979	69.9959	82.5314	38.9845
	+ tag style GT (comma)	22.9431	0.7143	0.2339	0.1805	0.4983	71.0330	0.5951	70.1340	82.8050	38.7870
	+ caption style GT	**23.0546**	**0.7182**	**0.2317**	**0.1797**	0.4944	70.8355	0.5932	**74.1048**	**85.1767**	**46.7137**
C	OSEDiff	**23.9596**	0.7533	**0.2147**	**0.1666**	0.4803	71.1559	0.5672	44.6204	58.4379	10.4372
	+ tag style GT (space)	23.9309	0.7531	0.2170	0.1679	0.4791	**71.1938**	**0.5718**	49.4762	64.8461	11.4528
	+ tag style GT (comma)	23.9462	0.7532	0.2166	0.1675	0.4760	71.0987	0.5716	49.7199	64.6750	12.7221
	+ caption style GT	23.9394	**0.7546**	0.2179	0.1695	**0.4811**	71.1110	0.5676	**51.4657**	**66.4082**	**14.4993**
E	TeReDiff	**23.5188**	**0.7233**	**0.2362**	**0.1796**	0.5638	70.5181	0.6183	51.2383	65.1083	16.9252
	+ tag style GT (space)	23.2760	0.7188	0.2393	0.1820	0.5608	70.7581	0.6166	55.3553	69.1448	21.2412
	+ tag style GT (comma)	23.3000	0.7200	0.2382	0.1814	0.5611	70.8026	0.6142	56.6301	70.0912	23.3286
	+ caption style GT	23.1785	0.7176	0.2388	0.1812	**0.5694**	**71.0449**	**0.6251**	**59.5128**	**72.7708**	**26.7137**
	UniT	22.7672	0.6503	0.2814	0.2072	**0.5159**	**68.7382**	**0.6315**	61.3886	76.4538	24.1185
	+ tag style GT (space)	23.0685	0.6699	0.2722	0.2004	0.5042	68.3784	0.6182	64.5635	78.7001	29.5910
	+ tag style GT (comma)	23.0727	0.6701	0.2721	0.2003	0.5039	68.3584	0.6181	64.4417	78.5063	29.6474
	+ caption style GT	**23.1021**	**0.6725**	**0.2694**	**0.1994**	0.5024	68.3612	0.6169	**70.6618**	**82.8848**	**40.4231**

B Additional Details on Reasoning Transfer via Captioning

B.1 Observation 1: Ground-Truth Transcripts Improve Text Restoration

This section provides two ablations on the GT transcript caption of Section 4.1, one on its format and one on adding a scene description.

Caption format ablation. Text-conditioned SR models differ in the caption format they are trained with. It is therefore unclear whether the GT transcripts should be injected in each model's training-time format or in a single unified format. We compare three formats that mirror the conventions of existing SR models, namely (1) tag style GT (space), which separates the transcripts with spaces [15] ("t_1 t_2 $\cdots$ t_n"), (2) tag style GT (comma), which separates them with commas [37] ("t_1, t_2, $\cdots$, t_n"), and (3) caption style GT, which lists them in a natural-language caption ("`The image with the text “`t_1`”, “`t_2`”, ⋯, “`t_n`”.`").

Table F shows that GT transcripts improve GTTCA over the original caption for every model, regardless of the format. Among the three formats, the natural-language caption achieves the best GTTCA for every model on both levels, even for UniT and OSEDiff, whose own conventions are tag style. We hypothesize that this is because the text-to-image models underlying these SR models were pretrained on natural-language captions, even though the SR models were later fine-tuned with various caption formats. These results support our choice in Section 4.1 to inject the GT transcripts as a natural-language caption.

Scene description ablation. While GT transcripts improve GTTCA, they lower NR-IQA scores for most models (Table 2). We hypothesize that this drop arises because a transcript-only caption departs from the captions these models are trained with, which mostly describe the scene. To test this, we prepend a one-sentence scene description of each LR image, generated by Gemma 4, to the caption-style GT. Table G shows that the scene description largely recovers NR-IQA. The scene description also changes GTTCA. It lowers overall GTTCA by 1.2 to 2.1 points for SUPIR, FaithDiff, OSEDiff, and TeReDiff, whereas it leaves DiT4SR nearly unchanged and improves UniT on both levels. We attribute this difference to the text encoders of the underlying text-to-image models.

Table G: **Adding a scene description to the caption-style GT largely recovers no-reference perceptual quality.** Each text-conditioned SR model is evaluated with its original caption, with the caption-style GT of its Level 1 and Level 2 instances, and with a Gemma 4 scene description prepended to that caption. Within each model block, **bold** marks the best value for each metric, and blue/red indicate improvement/decline over the original caption. The HR oracle row reports each metric on the HR images for reference. ↑(↓): higher (lower) is better.

Group	Model	Fidelity		Perceptual					Text Restoration		
				Reference		No Reference			GTTCA		
		PSNR-Y↑	SSIM-Y↑	LPIPS↓	DISTS↓	MANIQA↑	MUSIQ↑	CLIP-IQA↑	Overall↑	Level 1↑	Level 2↑
	HR (oracle)	80.0000	1.0000	0.0000	0.0000	0.4261	66.2367	0.5403	94.6813	96.2942	90.6911
B	SUPIR	24.2971	**0.7098**	0.2622	0.1947	0.4608	63.5998	0.5922	55.2984	68.5291	22.5670
	+ caption style GT	**24.5912**	0.7038	0.2732	0.1964	0.4349	60.9064	0.5714	**69.7686**	**81.6534**	**40.3667**
	+ scene + caption style GT	23.9171	0.7059	**0.2497**	**0.1828**	**0.5293**	**67.4045**	**0.6405**	67.9009	79.0878	40.2257
	FaithDiff	24.2038	0.7363	0.2061	0.1623	**0.5095**	**71.6540**	**0.5821**	52.0097	65.1653	19.4640
	+ caption style GT	**24.4677**	**0.7467**	**0.1962**	**0.1567**	0.4951	71.4650	0.5739	**62.4523**	**75.3250**	**30.6065**
	+ scene + caption style GT	24.2462	0.7393	0.2022	0.1593	0.5011	71.5270	0.5770	61.2830	74.3444	28.9704
	DiT4SR	22.6539	0.6933	0.2477	0.1830	0.4973	**70.8502**	0.6074	59.1555	73.1471	24.5416
	+ caption style GT	**23.0546**	**0.7182**	**0.2317**	**0.1797**	0.4944	70.8355	0.5932	74.1048	**85.1767**	46.7137
	+ scene + caption style GT	22.6967	0.6936	0.2481	0.1835	**0.5002**	70.7191	**0.6103**	**74.1941**	84.9031	**47.7010**
C	OSEDiff	**23.9596**	0.7533	**0.2147**	**0.1666**	0.4803	71.1559	0.5672	44.6204	58.4379	10.4372
	+ caption style GT	23.9394	**0.7546**	0.2179	0.1695	0.4811	71.1110	0.5676	**51.4657**	**66.4082**	**14.4993**
	+ scene + caption style GT	23.9081	0.7512	0.2167	0.1677	**0.4872**	**71.1714**	**0.5722**	49.3707	63.7856	13.7094
E	TeReDiff	**23.5188**	**0.7233**	**0.2362**	0.1796	0.5638	70.5181	0.6183	51.2383	65.1083	16.9252
	+ caption style GT	23.1785	0.7176	0.2388	0.1812	0.5694	**71.0449**	**0.6251**	**59.5128**	**72.7708**	**26.7137**
	+ scene + caption style GT	23.3463	0.7209	0.2369	**0.1792**	**0.5733**	70.8363	0.6207	57.3934	70.6842	24.5134
	UniT	22.7672	0.6503	0.2814	0.2072	**0.5159**	**68.7382**	**0.6315**	61.3886	76.4538	24.1185
	+ caption style GT	23.1021	0.6725	0.2694	0.1994	0.5024	68.3612	0.6169	70.6618	82.8848	40.4231
	+ scene + caption style GT	**23.1256**	**0.6742**	**0.2662**	**0.1975**	0.5044	68.4496	0.6193	**71.1815**	**83.0787**	**41.7489**

SUPIR, FaithDiff, OSEDiff, and TeReDiff build on models that use only CLIP text encoders, whose short context can truncate or dilute the transcripts that follow the scene description. DiT4SR and UniT build on the same backbone [10], which additionally uses a T5 text encoder that handles long captions. RTC includes the scene description despite this small cost in GTTCA, since it preserves perceptual quality. A text-to-image backbone that handles long captions may therefore be a good starting point for future TAISR.

B.2 Observation 2: MLLMs Can Approximate Ground-Truth Transcripts

Reason type	Image Context	Logical Patterns	Prior Knowledge
Text Recall	81.9	69.5	69.1

We further break down Gemma 4's Level 2 text recall by the annotated cue (288 image-context, 246 logical-pattern, and 414 prior-knowledge instances, where an instance can carry more than one cue). Recall is highest for image context (81.9%), where the cue is visible elsewhere in the image, and similar for logical patterns (69.5%) and prior knowledge (69.1%). If data contamination drove the performance, prior knowledge, the cue most tied to pre-training data, would likely be read best. Instead, it is read no better than logical patterns. Together with the analysis in Section 4.2, this suggests that Gemma 4's Level 2 recall comes from reasoning over the image rather than from memorization.

B.3 Method: Reasoning Transfer via Captioning

The reasoning prompt asks the MLLM for a two-sentence caption. The first sentence describes the scene, which avoids most of the NR-IQA drop observed with transcript-only captions (Appendix B.1). The second sentence lists the transcripts inferred from the LR image. Figure C illustrates the overall pipeline with an example caption.

C Additional Experiments

This section shows that RTC generalizes to existing TAISR benchmarks, is robust to imperfect captions, and remains effective on Level 3 text.

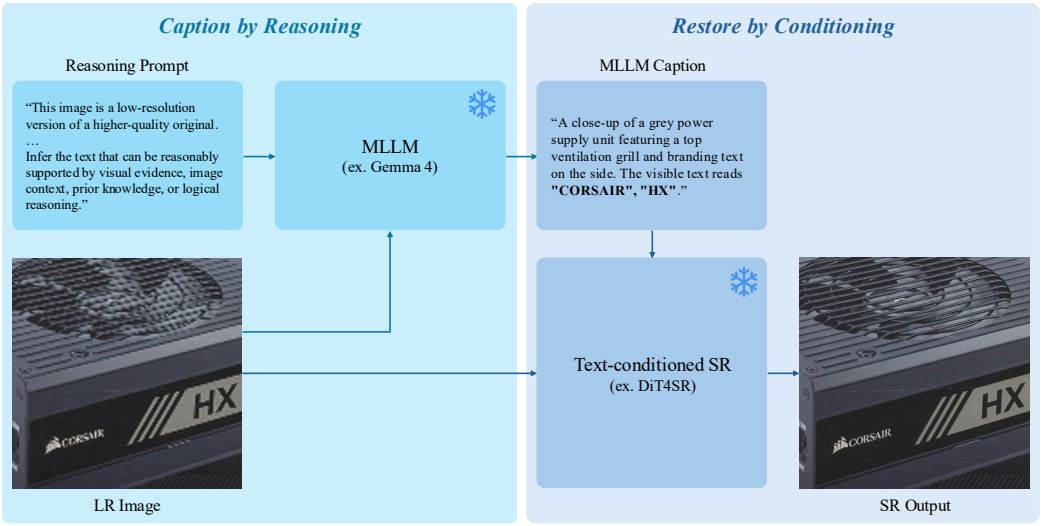


Figure C: **Overview of Reasoning Transfer via Captioning (RTC).** (1) Caption by reasoning: a frozen MLLM reads the LR image under the reasoning prompt and returns a two-sentence caption with a scene description and the inferred transcripts. (2) Restore by conditioning: a frozen text-conditioned SR model restores the LR image with this caption in place of its default caption.

Table H: **RTC improves text restoration on existing TAISR benchmarks.** Text-spotting metric (TESTR [46]) measured on RealText and the three degradation levels of SA-Text-test, comparing each text-conditioned SR model with and without RTC. The levels of SA-Text-test denote degradation strengths and are unrelated to the difficulty levels of ReasonText. Blue marks an improvement over the baseline. The HR oracle row reports the score on the HR images for reference.

Group	Model	RealText	SA-Text-test		
			Level 1	Level 2	Level 3
HR (oracle)		75.4443	76.2919	76.2919	76.2919
B	SUPIR	41.6263	23.5721	20.2629	12.2847
	+ RTC	50.9962	34.0435	28.4225	18.5857
	FaithDiff	43.4572	23.5721	20.7162	13.3726
	+ RTC	49.9731	28.4678	26.5186	18.7670
	DiT4SR	46.9575	25.1133	21.9855	15.2765
	+ RTC	56.4890	36.4461	31.6410	21.5775
C	OSEDiff	41.4647	19.6283	16.9991	10.9248
	+ RTC	46.0420	24.0254	20.8069	15.1405
E	TeReDiff	46.2036	23.3908	21.5775	15.5938
	+ RTC	49.4346	26.9719	24.4787	17.2711
	UniT	47.0113	26.2919	23.5267	16.7724
	+ RTC	54.8196	34.0888	30.3717	20.2629

RTC also improves text restoration on existing TAISR benchmarks. We further evaluate RTC on the two existing TAISR benchmarks, SA-Text-test and RealText. SA-Text-test consists of 1,000 LR–HR pairs at three synthetic degradation strengths, denoted Levels 1-3 in Table H, where a higher level means stronger degradation. These levels are unrelated to the difficulty levels of ReasonText. RealText consists of 847 LR–HR pairs of scene images with real-world degradations. Since both benchmarks are annotated by fully automatic pipelines without human verification, their labels contain non-trivial noise (Section 3.1). This noise would directly affect GTTCA, which relies on the GT polygons and transcripts. We therefore evaluate each model with and without RTC under the original text spotting protocol of these benchmarks.

Table I: **RTC is robust to caption errors.** (a) Level 2 GTTCA before (base) and after RTC (+RTC), split by whether the RTC caption contains the correct GT transcript. RTC improves GTTCA for all six SR models even when Gemma 4 is wrong. (b) The Gemma wrong subset pooled over the six SR models and split by how far the caption is from the GT transcript. The gain concentrates on near-miss captions that differ by at most two letters, from which the SR models can still restore the exact text. Blue marks an improvement over the baseline.

(a) Per-model, split by caption correctness

Group	Model	Gemma correct (n=499)		Gemma wrong (n=210)	
		base	+RTC	base	+RTC
B	SUPIR	25.7	46.5	15.2	20.5
	FaithDiff	21.2	33.1	15.2	19.0
	DiT4SR	26.9	55.3	19.0	23.3
C	OSEDiff	11.8	13.4	7.1	11.0
E	TeReDiff	18.8	25.9	12.4	17.1
	UniT	28.1	51.1	14.8	21.4

(b) Gemma wrong subset, by caption distance

Wrong L2 caption	base	+RTC	Δ
near-miss (≤2 letters)	17.8	25.9	+8.1
far (>2 letters)	10.2	11.6	+1.4

Table J: **RTC is effective at every difficulty level, including Level 3.** (a) GTTCA before (base) and after RTC (+RTC) on each difficulty level. RTC improves all six SR models on Level 1, Level 2, and even Level 3, though the Level 3 gains are smaller on average. Mean Δ is the average gain over the six models. (b) Text recall of Gemma 4 with the reasoning prompt on each level. Blue marks an improvement over the baseline.

(a) GTTCA per backbone and difficulty level

Group	Model	Level 1		Level 2		Level 3	
		base	+RTC	base	+RTC	base	+RTC
B	SUPIR	68.5	78.8	22.6	38.8	4.9	11.8
	FaithDiff	65.2	74.0	19.5	28.9	2.7	5.8
	DiT4SR	73.1	84.8	24.5	45.8	6.2	18.3
C	OSEDiff	58.4	64.3	10.4	12.7	1.4	2.0
E	TeReDiff	65.1	70.1	16.9	23.3	2.6	5.0
	UniT	76.5	82.6	24.1	42.3	5.1	14.1
Mean Δ		8.0		12.3		5.7	

(b) Gemma 4 text recall by level

Level 1	Level 2	Level 3
86.7	70.4	40.2

Table H reports the results on RealText and on the three degradation levels of SA-Text-test. In all four settings, RTC improves all six text-conditioned SR models without exception, indicating that its gains are not an artifact of a particular benchmark or metric. Moreover, UniT outperforms DiT4SR in every setting without RTC, whereas DiT4SR + RTC outperforms UniT + RTC in every setting. This reproduces the reversal observed on ReasonText (Section 5.2) and again suggests that reasoning should be a core axis of text-aware design.

RTC is robust to imperfect captions. Since RTC feeds the SR model a caption predicted by an MLLM, one might worry that a wrong prediction would degrade restoration. To examine this, we split the 709 Level 2 instances by whether the RTC caption contains the correct GT transcript, into a Gemma correct subset (499 instances) and a Gemma wrong subset (210 instances), and measure Level 2 GTTCA on each subset before and after RTC (Table I(a)). Even on the Gemma wrong subset, RTC improves GTTCA for all six SR models. Half of the wrong captions are near-misses that differ from the GT transcript by at most two letters, and the gain concentrates on them (17.8% to 25.9%), since the SR models can still restore the exact text from a near-correct caption together with the LR image (Table I(b)). For captions farther from the GT transcript, RTC still does not lower GTTCA on average (10.2% to 11.6%). RTC is therefore robust to imperfect captions, which matters in practice since the MLLM may misread degraded text.

RTC remains effective on Level 3 text. ReasonText's Level 3 comprises text that none of our annotators could read even from the full LR patch, so it is excluded from our main evaluation. Nonetheless, since RTC injects an MLLM caption that may be wrong, a natural concern is that it might harm this hardest case. Table J (a) shows that it does not. RTC improves GTTCA on Level 3 for all six SR models, though by less than on the other levels on average. These gains are possible because Gemma 4 with the reasoning prompt still recalls 40.2% of Level 3 text (Table J(b)). We attribute this to how Level 3 is defined. Level 3 means that none of our annotators could read the text, not that the text is intrinsically unrecoverable. This set is therefore heterogeneous. Some instances, such as serial numbers, are genuinely unrecoverable, while others might have been read by a larger annotator pool or by domain experts with the relevant prior knowledge. Gemma 4 likely reads some instances of the latter kind, reflecting the limitation of our annotation protocol noted in Section 6.

D Limitations and Broader Impact

Limitations. We summarize the limitations discussed in Section 6. First, ReasonText is human-curated and built on existing real-world SR benchmarks, so it is limited in size and scope, with 513 LR–HR pairs of natural images and only alphanumeric text annotated. Second, even the best model with RTC remains far below the HR oracle on Level 2 (45.84% versus 90.69%). Since the same model reaches at most 47.70% even with GT transcripts (Table G), most of this gap lies not in reading the text but in rendering it, and a stronger MLLM alone would bring limited gains. Third, RTC applies only to text-conditioned SR models and cannot be used with SR models that have no text channel.

Broader Impact. By enabling SR models to recover text that was previously unreadable from local pixels alone, this work has the potential to benefit a wide range of socially valuable applications. Improved restoration of degraded text can (i) support accessibility for users with low vision by making blurred signs, labels, and documents legible, (ii) aid cultural and historical preservation by helping recover text in old photographs, archival scans, and damaged manuscripts, and (iii) contribute to public safety and navigation through clearer recovery of road signs, transit information, and emergency notices captured under poor imaging conditions. More broadly, our findings suggest that reasoning, rather than purely local perception, is a productive direction for restoration tasks, encouraging the community to explore restoration pipelines that integrate world knowledge and contextual understanding. At the same time, text restored through reasoning is a plausible inference rather than evidence, and it may occasionally diverge from the true content. Such outputs should not be treated as ground truth in high-stakes settings such as forensic analysis or document verification, and the ability to read degraded text calls for care with images containing private information, such as license plates or personal documents. The decoupled design of RTC helps mitigate these risks, since the MLLM caption exposes the inferred text and makes it possible to trace which component caused an error. Overall, we hope ReasonText, GTTCA, and RTC together provide a constructive foundation for building TAISR systems that are not only sharper, but also more thoughtful about the text they restore.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions stated in the abstract and introduction (ReasonText, GTTCA, and RTC) are directly supported by the experiments in Section 3, 4, and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a discussion of limitations in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results such as formal theorems or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the construction details of the ReasonText benchmark (Section 3.1, Appendix A.1), the exact computation procedure of GTTCA (Section 3.2, Appendix A.2), and the full RTC pipeline including the prompt and caption template (Section 4.3, Appendix B.3).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code and the dataset will be provided upon publication to support validation of the reported results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Evaluation protocols (Section 3) and the RTC pipeline including the MLLM, prompt, and caption template (Section 4) are fully specified. All SR models are run off-the-shelf with their default configurations; RTC is training-free.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: All metrics are deterministic and evaluated on a fixed benchmark with off-the-shelf SR models and we fix the seed of MLLM sampling, so run-to-run variance is negligible. The consistency of our findings across twelve models (Table 1) and six SR models (Table 4) supports the robustness of the claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are described in Seciton 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a discussion of limitations in Appendix.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve the release of data or models that pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites the original works for all datasets and models used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the ReasonText benchmark, with construction details, annotation protocol, and statistics documented in Section 3.1 and Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: An MLLM (Gemma 4) is a core component of our method RTC, serving as the reasoning module that infers degraded text from LR images and supplies it to SR models via a caption (Section 4).

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.