

CARE What Fails: Contrastive Anchored-REflection for Verifiable Multimodal Reasoning

Supplementary Material

6. Notation

Sym.	Meaning
x	Multimodal input instance.
\mathbf{I}	Image or set of images in the input.
\mathbf{q}	Text query in the input.
π_θ	Trainable policy that generates rationale and answer.
\hat{r}	Rationale inside the <think> span.
\hat{y}	Final answer inside the <answer> span.
\mathcal{V}	Programmatic verifier of answers.
acc	Verifier accuracy signal.
fmt	Verifier format-compliance signal.
r	Overall reward signal.
r_i	Reward assigned to the i -th rollout.
y	A model response.
y_i	The i -th rollout for a given input.
\mathcal{Y}	Collection of rollouts for a given input.
G	Number of rollouts sampled per input.
T_i	Total token count of the i -th response.
T_i^{ans}	Token count inside the answer span of the i -th response.
T_i^{think}	Token count inside the rationale span of the i -th response.
\mathcal{P}	Indices of verifier-positive rollouts.
\mathcal{Y}^+	Anchor rollout selected from verifier-positive rollouts.
\mathcal{F}	Failure pool of verifier-negative rollouts.
\mathcal{N}	Selected hard negatives.
K	Target number of negatives in the subgroup.
K'	Realized number of negatives in the subgroup.
S	Anchored subgroup used for normalization and updates.
S^*	Indices with nonzero sequence-level advantages.
M	Preselection size before diversity pruning among negatives.
\tilde{h}_i	Normalized rationale embedding of the i -th rollout.
\tilde{h}_+	Normalized rationale embedding of the anchor.
d_{cos}	Cosine-distance metric in the embedding space.
μ_S	Mean reward within the subgroup.
σ_S	Standard-deviation scale within the subgroup.
$A[y]$	Sequence-level advantage for a rollout.
s	Scaling factor applied to negative advantages.
s_{refl}	Reduced scaling factor for failed reflection samples.
$\rho_{i,t}$	Importance ratio at token position t for the i -th rollout.
$w_{i,t}$	Region weight at token position t for the i -th rollout.
γ^+	Weight for rationale tokens of verifier-positive samples.
$a_{i,t}$	Per-token advantage at position t for the i -th rollout.
ϵ	Small constant for numerical stabilization.
ϵ_w	Small constant used in token-level normalization.
ϵ_{low}	Lower clipping threshold in the surrogate objective.
ϵ_{high}	Upper clipping threshold in the surrogate objective.
π_{old}	Behavior policy used for importance ratios.
π_{ref}	Reference policy used in KL regularization.
β	Coefficient of the KL regularizer.
y^-	Hard negative used for reflection-guided resampling.
RGR	Reflection-Guided Resampling procedure.
δ	Magnitude used in the all-negative rescue pseudo-contrast.
t	Pseudo-anchor used in the all-negative rescue.
r^+	Reward of the anchor rollout.
m_-	Mean reward among negatives in the subgroup.
Δ	Reward gap between the anchor and negatives.
Var_-	Reward variance among negatives in the subgroup.
α	Attenuation factor in the mechanistic analysis.

Table 3. **Notation.** Single-symbol entries with descriptions that avoid formulas.

7. Proof of K Mechanistic signature

Setup. Let the anchored subgroup be $S = \{y^+\} \cup \{y_1^-, \dots, y_{K'}^-\}$. The anchor has reward r^+ ; negatives are i.i.d. with mean m_- and variance Var_- . Define $\Delta = r^+ - m_- > 0$. Advantages are group-normalized as in Eqs. (5) to (7) (we drop the tiny ϵ for clarity).

Proposition 1 (Two-level K -signature under small dispersion). *Under the assumptions above, a plug-in (delta-method) approximation that replaces the random denominator by $\sqrt{\mathbb{E}[\sigma_S^2]}$ yields*

$$\mathbb{E} A[y^+] \approx \alpha_{K'} \sqrt{K'}, \quad \mathbb{E} \bar{A}[y^-] \approx -\alpha_{K'} \frac{1}{\sqrt{K'}},$$

where

$$\alpha_{K'} = \frac{\Delta}{\sqrt{K' \text{Var}_- + \Delta^2}} \in (0, 1].$$

Sketch. We have $\mathbb{E}[\mu_S] = m_- + \frac{\Delta}{K'+1}$ and

$$\begin{aligned} \mathbb{E}[(r^+ - \mu_S)^2] &= \frac{K'^2}{(K'+1)^2} \Delta^2 \\ &\quad + \frac{K'}{(K'+1)^2} \text{Var}_-. \end{aligned}$$

Similarly,

$$\begin{aligned} \mathbb{E}[(r_j^- - \mu_S)^2] &= \text{Var}_- \left(1 - \frac{2}{K'+1} + \frac{K'}{(K'+1)^2} \right) \\ &\quad + \frac{\Delta^2}{(K'+1)^2}. \end{aligned}$$

Therefore

$$\begin{aligned} \mathbb{E}[\sigma_S^2] &= \frac{1}{K'+1} \left(\mathbb{E}[(r^+ - \mu_S)^2] + K' \mathbb{E}[(r_j^- - \mu_S)^2] \right) \\ &= \frac{K'^2}{(K'+1)^2} \text{Var}_- + \frac{K'}{(K'+1)^2} \Delta^2. \end{aligned}$$

Also $\mathbb{E}[r^+ - \mu_S] = \frac{K'}{K'+1} \Delta$. Approximating $\mathbb{E}[(r^+ - \mu_S)/\sigma_S] \approx \mathbb{E}[r^+ - \mu_S]/\sqrt{\mathbb{E}[\sigma_S^2]}$ gives

$$\begin{aligned} \mathbb{E} A[y^+] &\approx \frac{\Delta \frac{K'}{K'+1}}{\sqrt{\frac{K'^2}{(K'+1)^2} \text{Var}_- + \frac{K'}{(K'+1)^2} \Delta^2}} \\ &= \frac{\Delta \sqrt{K'}}{\sqrt{K' \text{Var}_- + \Delta^2}} = \alpha_{K'} \sqrt{K'}. \end{aligned}$$

For a negative y_j^- , $\mathbb{E}[r_j^- - \mu_S] = -\frac{\Delta}{K'+1}$, which yields the stated $\mathbb{E} \bar{A}[y^-]$. \square

Corollary 1 (Binary rewards). *If $r^+ = 1$ and all negatives satisfy $r_j^- = 0$ (i.e., $\text{Var}_- = 0$, $\Delta = 1$), then the z -scored advantages are exact:*

$$A[y^+] = \sqrt{K'}, \quad A[y_j^-] = -\frac{1}{\sqrt{K'}}$$

matching Eq. (10).

Corollary 2 (All-negative Rescue). *Under the pseudo-reward assignment in Eq. (14) on $S = \{t\} \cup \mathcal{N}$, with $r'[t] = \delta$, $r'[j] = -\delta/K'$ for $j \in \mathcal{N}$, we have $\mu_S = 0$ and*

$$\sigma_S^2 = \frac{\delta^2}{K'}$$

Hence

$$A[t] = \sqrt{K'}, \quad A[j] = -\frac{1}{\sqrt{K'}}$$

Thus the rescue update reproduces the $\sqrt{K'}$ signature exactly.

8. Implementation Details

8.1. Datasets Details

We build our visual reasoning training set by pooling three publicly available sources—ChartQA [26], Geometry3K [23], and ViRL39K [39]. All examples are standardized into a unified image–prompt–answer schema, followed by light cleaning and cross-corpus deduplication. The merged collection contains approximately **49.3K** unique multimodal prompts. Table 4 summarizes the per-source counts and shares prior to merging: **7,398** from ChartQA (15.3%), **2,101** from Geometry3K (4.3%), and **38,870** from ViRL39K (80.4%), totaling **48,369** items. This mixture provides complementary coverage of chart reasoning, geometric diagram understanding, and broader visual reasoning.

Table 4. Sources and sizes of the visual reasoning training set.

Dataset	Reference	Count	Share
ChartQA	[26]	7,398	15.3%
Geometry3K	[23]	2,101	4.3%
ViRL39K	[39]	38,870	80.4%
Total		48,369	100.0%

8.2. Experimental Details

Sampling and groups. Unless otherwise noted, we sample $G=8$ rollouts per prompt for all methods. For CARE, we form an anchor-aware hard-negative subgroup of size $K=4$ (Sections 3.2 and 3.3). In all main results we set $M = 6$. Within each subgroup we compute group-normalized advantages (Section 3.4); negatives are

down-weighted by the factor $s=0.5$ (Eq. (8)). When a verifier-positive anchor exists, Reflection-Guided Resampling (RGR; Section 3.6) performs one training-only resample on the selected hard negative; reflected failures are further reduced by $s_{\text{refl}}=s/2$. In all-negative groups we enable the *Rescue* mechanism with pseudo-contrast magnitude $\delta=0.1$ (Section 3.7). Region weights follow Equation (11) with $\gamma^+=0.005$ for positive <think> (note: <reasoning> with Qwen3VL-8B) tokens and 0 on negatives; answer tokens always receive weight 1.

Optimization. All methods use the same PPO-style clipped objective (Equation (12)) with a fixed KL regularizer to a reference policy (Equation (13)). Decoding hyperparameters (e.g., temperature, nucleus/top- p) are held constant across methods and reused at evaluation; inference always uses a *single* decode and *no* reflection.

Baselines and fairness. We re-train GRPO [33], DAPO [49], and GSPO [56] on the same 49.3K mixture with the same rollout budget $G=8$, identical decoding, and a matched optimization budget (steps, tokens, and KL schedule). Baselines keep their standard token-uniform weighting (no region weighting) and do not use reflection. All models—ours and baselines—are trained from the same SFT cold-start and use the same verifier signals *acc* and *fmt* combined by Eq. (1).

8.3. Evaluation Details

Common protocol. All evaluations use a *single* decode per example under identical decoding hyperparameters for every model. We extract the text inside the <answer>...</answer> span and compute *exact match* against the benchmark’s key using the official normalization rules (no custom post-processing). Scores are reported per benchmark; the *macro average* is the unweighted mean across all six benchmarks. The open-ended answers of benchmarks are judged by GPT-5.

MathVista_{mini} [24] targets mathematical and scientific reasoning with plots, charts, and diagrams; answers are short and programmatically verifiable.

MathVerse_{mini} [55] covers math-intensive visual problems (algebra/geometry/physics) presented as figures or rendered text-in-image.

MATH-Vision_{full} [40] evaluates multi-step math reasoning grounded in images (e.g., annotated diagrams); answers are discrete and exact-matchable.

MMMU_{val} [50] spans multi-disciplinary, multi-image questions; we report accuracy on the public validation set.

MMMU-Pro_{standard} and **MMMU-Pro_{vision}** [51] evaluate professional-level reasoning, with the *vision* split focusing more heavily on visual content. Both provide deterministic answer keys for exact-match evaluation.

Algorithm 1 CARE: Anchored-Contrastive Objective with Reflection-Guided Resampling (updated)

Require: input x ; policy π_θ ; old policy π_{old} ; reference π_{ref} ; rollout budget G ; target subgroup size K ; neg_scale s ; reduced neg_scale $s_{\text{reff}} \leq s$; answer/rationale weight γ^+ ; reward mix $\lambda \in [0, 1]$; clipping ($\epsilon_{\text{low}}, \epsilon_{\text{high}}$); KL coef β ; normalization constants ϵ, ϵ_w ; rescue magnitude δ

- 1: Sample G rollouts $\mathcal{Y} = \{y_i\}_{i=1}^G$; parse `<think>` and `<answer>` spans; record lengths $T_i^{\text{think}}, T_i^{\text{ans}}$
- 2: For each y_i : compute verifier signals $(\text{acc}_i, \text{fmt}_i) \leftarrow \mathcal{V}(\text{extract_answer}(y_i))$; set $r_i \leftarrow (1 - \lambda) \text{acc}_i + \lambda \text{fmt}_i$
- 3: Define $\mathcal{P} \leftarrow \{i : \text{acc}_i = 1\}$ (positives), $\mathcal{F} \leftarrow \{i : \text{acc}_i = 0\}$ (failures)
- 4: **if** $\mathcal{P} \neq \emptyset$ **then** ▷ non-all-negative group
- 5: $a \leftarrow y^+ \leftarrow \arg \min_{i \in \mathcal{P}} (T_i^{\text{think}}, T_i^{\text{ans}})$ ▷ anchor = shortest `<think>`, tie-break shorter answer
- 6: $K' \leftarrow \min(K, |\mathcal{F}|)$
- 7: **if** $K' = 0$ **then skip update and continue** ▷ all-positive → avoid arbitrary drift
- 8: **end if**
- 9: $\mathcal{N} \leftarrow \text{AnchorAwareNegatives}(\mathcal{F}, K', y^+)$ ▷ rank failures by cosine to anchor’s `<think>` embedding; stop-gradient; brief farthest-first de-dup
- 10: $S \leftarrow \{y^+\} \cup \{y_1^-, \dots, y_{K'}^-\}$; $\tilde{r}(y) \leftarrow r(y)$
- 11: **if** reflection enabled **then** ▷ Reflection-Guided Resampling (train-time only)
- 12: Choose $y^{\text{neg}} \in \mathcal{N}$ (e.g., hardest by d_{cos})
- 13: Insert concise repair cue into `<think>` of y^{neg} ; decode once to obtain y^{rgr}
- 14: Parse y^{rgr} ; compute $(\text{acc}_{\text{rgr}}, \text{fmt}_{\text{rgr}}), r_{\text{rgr}} \leftarrow (1 - \lambda) \text{acc}_{\text{rgr}} + \lambda \text{fmt}_{\text{rgr}}$
- 15: **if** $\text{acc}_{\text{rgr}} = 1$ **then** ▷ success defined by verifier-positive
- 16: $S \leftarrow (S \setminus \{y^{\text{neg}}\}) \cup \{y^{\text{rgr}}\}$; $\tilde{r}(y^{\text{rgr}}) \leftarrow r_{\text{rgr}}$
- 17: **else** ▷ reflection fails
- 18: $S \leftarrow S \cup \{y^{\text{rgr}}\}$; $\tilde{r}(y^{\text{rgr}}) \leftarrow r_{\text{rgr}}$; mark y^{rgr} for reduced scaling s_{reff}
- 19: **end if**
- 20: **end if**
- 21: **else** ▷ all-negative group
- 22: $t \leftarrow \text{ProxyAnchor}(\mathcal{F}) \equiv \arg \max_{i \in \mathcal{F}} \log \pi_{\text{old}}(y_i | x)$
- 23: $K' \leftarrow \min(K, |\mathcal{F}|)$; $\mathcal{N} \leftarrow \text{AnchorAwareNegatives}(\mathcal{F} \setminus \{t\}, K', t)$
- 24: $S \leftarrow \{t\} \cup \mathcal{N}$; assign pseudo-rewards on S : $\tilde{r}(t) \leftarrow \delta, \tilde{r}(j) \leftarrow -\delta/K'$ for $j \in \mathcal{N}$; $a \leftarrow t$
- 25: **end if**
- 26: $K_S \leftarrow |\{y \in S \setminus \{a\} : \tilde{r}(y) < \tilde{r}(a)\}|$ ▷ negatives actually present after any reflection
- 27: **if** $K_S = 0$ **then skip update and continue**
- 28: **end if**
- 29: $\mu_S \leftarrow |S|^{-1} \sum_{y \in S} \tilde{r}(y)$; $\sigma_S \leftarrow \sqrt{|S|^{-1} \sum_{y \in S} (\tilde{r}(y) - \mu_S)^2} + \epsilon$
- 30: Initialize $A(y) \leftarrow 0$ for all $y \in \mathcal{Y}$ (and any reflected sample)
- 31: **for** each $y \in S$ **do**
- 32: $A_{\text{raw}}(y) \leftarrow (\tilde{r}(y) - \mu_S) / \sigma_S$
- 33: **end for**
- 34: Set $A(a) \leftarrow A_{\text{raw}}(a)$
- 35: **for** each $y \in S \setminus \{a\}$ **do**
- 36: $s_y \leftarrow s$; **if** y is a failed reflected sample **then** $s_y \leftarrow s_{\text{reff}}$
- 37: $A(y) \leftarrow -s_y |A_{\text{raw}}(y)|$
- 38: **end for**
- 39: **Equalize update size:** **if** $K_S < K$ **then** $A(\cdot) \leftarrow \sqrt{K/K_S} A(\cdot)$
- 40: **Region weights:** for token t in y_i :
 $w_{i,t} = 1$ if $t \in \langle \text{answer} \rangle$; $w_{i,t} = \gamma^+$ if $t \in \langle \text{think} \rangle$ and $\text{acc}_i = 1$; $w_{i,t} = 0$ if $t \in \langle \text{think} \rangle$ and $\text{acc}_i = 0$
- 41: Per-token advantage $a_{i,t} \leftarrow A[y_i] \cdot \frac{w_{i,t}}{\sum_{u=1}^{T_i} w_{i,u} + \epsilon_w}$
- 42: PPO loss with clipping:
 $\mathcal{L}_{\text{PG}} = -\frac{1}{|S^*|} \sum_{i \in S^*} \sum_t \min\{\rho_{i,t} a_{i,t}, \text{clip}(\rho_{i,t}, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}) a_{i,t}\}$, where $S^* = \{i : A[y_i] \neq 0\}$ and $\rho_{i,t} = \exp(\log \pi_\theta - \log \pi_{\text{old}})$
- 43: Add KL regularizer: $\mathcal{L} \leftarrow \mathcal{L}_{\text{PG}} + \beta \text{KL}[\pi_\theta \| \pi_{\text{ref}}]$; update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

9. Additional Ablations

9.1. Ablation on K and Equalization

We record **Val-Acc** and **KL/step** every 100 training steps. Figures Figures 9 and 10 depict the trajectories. All con-

Table 5. Training-time configuration (shared vs. method-specific). “Uniform” = token-uniform weighting; “RW” = region-weighted scheme in Eq. (11).

Method	G	K	Neg. scale	Rescue	Reflection (train)	Token weights	Clip _{low}	Clip _{high}	KL coef β
CARE (ours)	8	4	$s=0.5$	$\delta=0.1$	Yes (1 resample)	RW ($\gamma^+=0.005$)	0.20	0.28	0.02
CARE w/o RGR	8	4	$s=0.5$	$\delta=0.1$	No	RW ($\gamma^+=0.005$)	0.20	0.28	0.02
GRPO	8	—	—	—	No	Uniform	0.20	0.20	0.02
DAPO	8	—	—	—	No	Uniform	0.20	0.28	-
GSPO	8	—	—	—	No	Uniform	3e-4	4e-4	-

Shared across all methods unless noted: same data, verifier, and decoding.

figurations start near $\approx 17\%$ accuracy and differ only in the subgroup size $K \in \{2, 4, 6\}$ and whether update-size equalization $\sqrt{K/K'}$ (Eq) is enabled. Across the board, enabling Eq accelerates accuracy growth and yields higher final performance while simultaneously lowering and stabilizing KL. The strongest setting is $K=4$ with Eq enabled, converging at $\approx 61.5\%$ versus 59.8% without Eq; similar but slightly smaller gains appear for $K=2$ (57.8% vs. 56.8%) and $K=6$ (60.3% vs. 58.5%). The KL trajectories mirror these patterns: for $K=4$, the final KL drops from ≈ 0.70 (Eq off) to ≈ 0.48 (Eq on); for $K=6$, from ≈ 0.72 to ≈ 0.50 ; and for $K=2$, from ≈ 0.75 to ≈ 0.50 . We also observe narrower confidence intervals with Eq on throughout most of training, indicating reduced seed-to-seed variance and fewer regressions. These results are consistent with the mechanism that the realized hard-negative count K' often falls below K : without equalization the effective update magnitude shrinks whenever $K' < K$, slowing progress and increasing volatility, whereas the $\sqrt{K/K'}$ factor restores a comparable update scale and produces steadier policy updates, reflected in both faster accuracy gains and smoother, lower KL. Taken together, the Acc–KL trade-off favors $K=4$ with Eq on as a robust default; larger K remains competitive provided Eq is enabled, while smaller K is viable under tight budgets but saturates lower.

9.2. Ablation Analysis on Anchor

Holding accuracy approximately constant between the two anchor rules, we still observe a clear separation in reasoning length: with the same initial average think length at initialization, the *Shortest-Think* anchor shortens the average think tokens markedly faster than a *Random* anchor (Figure 11). By the end of training the gap stabilizes at roughly 15–20 tokens, indicating that the model produces more compact rationales without sacrificing task accuracy. This pattern is consistent with our mechanism: choosing a concise, verifier-positive anchor concentrates the contrast on near-miss negatives and reduces credit dilution in the rationale region, yielding cleaner reasoning at comparable accuracy. Taken together, these results demonstrate that the

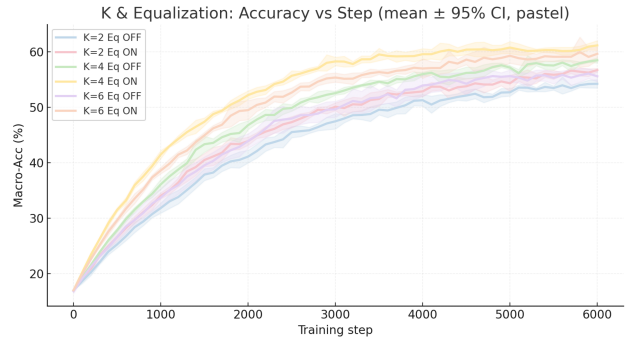


Figure 9. Val-Acc every 100 steps for $K \in \{2, 4, 6\}$ with/without equalization.

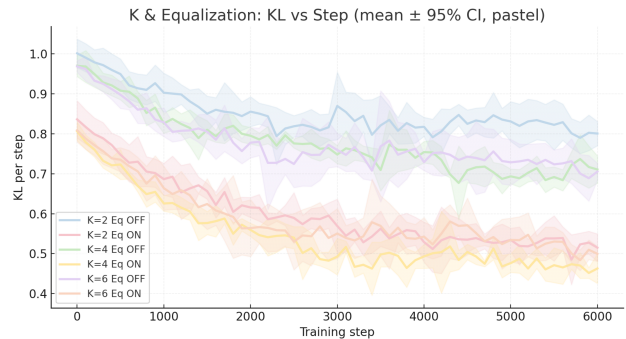


Figure 10. KL per 100 steps.

anchoring rule controls how efficiently learning signal is allocated to the rationale; *Shortest-Think* achieves shorter, more focused reasoning while maintaining the same accuracy as *Random*.

9.3. Ablation Analysis on s reflection

As shown in Figure 12, across training, we observe a consistent ordering: $s_{\text{ref}}=s/2$ yields the largest per-trigger gain, $3s/4$ is close but lower, while $s/4$ and s lag behind. Early in training (≈ 0 – $2k$ steps) the $s/2$ curve rises faster; mid-training ($\approx 2k$ – $4k$) its advantage persists with narrower con-

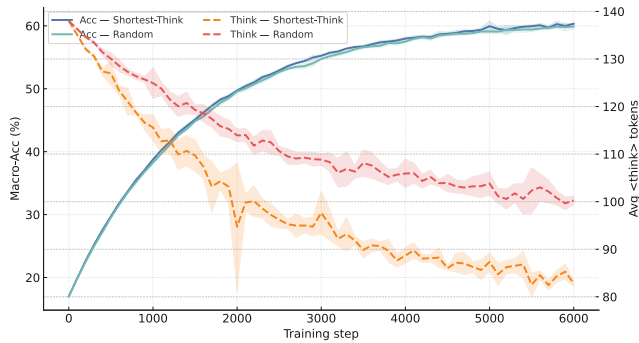


Figure 11. Val-Accuracy and Thinking token length of *Shortest-Think* and *Random* Anchor selection.

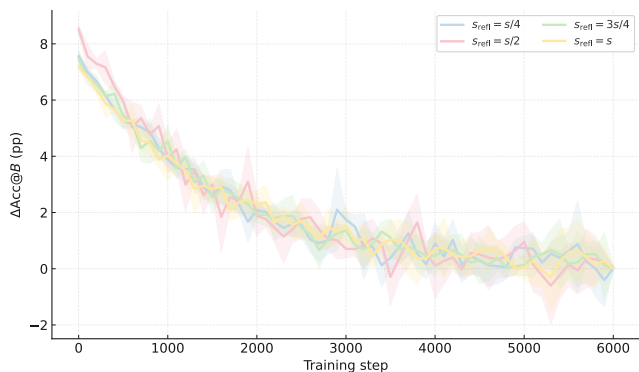


Figure 12. Across training, ACC@300 on different s_{ref} scale.

fidence bands; late in training all curves flatten but $s/2$ remains on top. This pattern matches the mechanism-level intuition: too small a scaling ($s/4$) under-penalizes failed reflections and does not sufficiently move the policy off the error mode; too large a scaling (s) overshoots and increases brittleness; a moderate scaling ($s/2$) balances gradient magnitude and exploration radius, keeping updates within a repairable neighborhood and converting more hard errors into measurable short-horizon accuracy. We therefore adopt $s_{\text{ref}}=s/2$ as the default choice.

10. Future Directions and Limitations

10.1. Limitations

While CARE demonstrates strong robustness on verifiable reasoning tasks, its current reliance on binary correctness signals restricts its primary application to domains with objective ground truths, such as mathematics and coding, leaving the extension to open-ended generation contingent on the development of reliable reward models. Additionally, although the all-negative rescue mechanism effectively mitigates gradient collapse in sparse-reward scenarios, the framework operates most efficiently when the policy can discover at least one successful rollout to anchor the con-

trastive update, implying that extremely hard tasks with near-zero pass rates may still benefit from initial warm-up supervision. Finally, our heuristic preference for the shortest successful rationale is a deliberate design choice to optimize inference efficiency; while this successfully curbs verbosity, it entails a trade-off that may theoretically discourage extended reasoning chains in specific edge cases where exhaustive self-verification is strictly necessary.

10.2. Future Directions

Step-level verification and credit. Augment the reward with *intermediate* verifiers that check sub-goals (e.g., variable extraction, unit conversion, geometric equivalences), enabling token- or span-level advantages on `<think>` without relying on a global scalar.

Multi-anchor or anchor ensembles. Replace the single anchor with a *set* of diverse positives (e.g., by clustering successful rationales) and distribute positive credit across them. This can mitigate brevity bias, preserve solution diversity, and improve stability on multi-modal prompts with several correct styles.

Learned negative selection. Move beyond cosine heuristics by training a selector that predicts “fixability” or causal proximity to the anchor (e.g., via contrastive learning on success/failure edits). Hybrid selectors could balance semantic proximity with coverage of distinct error families.

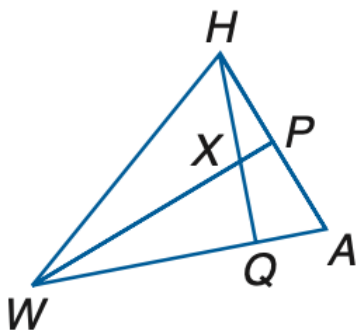
Adaptive subgroup sizing and scaling. Learn K' and the negative scaling s from data (or schedule them dynamically) to maintain a target signal-to-noise ratio inside the subgroup. This could reduce sensitivity to class imbalance and reward variance across prompts.

11. Quantitative Case Studies

We present detailed case studies illustrating the components of the CARE training objective. For each example, we show:

1. **The Anchor (y^+):** The shortest successful rollout used as the positive reference.
2. **Hard Negatives (y^-):** Failed rollouts selected by cosine similarity. Note how the reasoning traces are semantically close to the anchor (e.g., correct formulas or data extraction steps) but fail due to specific hallucinations or omissions.
3. **RGR Success:** A sample where the model, prompted with the repair cue, successfully corrects a hard negative.

Case 1: Geometry



Question: If \overline{WP} is a perpendicular bisector, $m\angle WHA = 8q + 17$, $m\angle HWP = 10 + q$, $AP = 6r + 4$, and $PH = 22 + 3r$, find q .

GT Answer: 7

1. Anchor (y^+) [Positive]

Shortest correct reasoning path.

$\overline{WP} \perp \overline{HA} \implies \angle WPH = 90^\circ$. In $\triangle WHP$, angles sum to 180. Thus $(8q + 17) + (10 + q) + 90 = 180$.

$9q + 117 = 180 \implies 9q = 63 \implies q = 7$.

Answer: 7

2. Hard Negatives (y^-)

Reasoning is semantically close (uses geometric properties) but hallucinates constraints.

Neg A (Isosceles Hallucination): $\triangle WHA$ is isosceles. Base angles equal: $8q + 17 = 10 + q \implies 7q = -7 \implies q = -1$.

Neg B (Summation Error): Sum of given angles is 180 (ignoring right angle). $(8q + 17) + (10 + q) = 180 \implies 9q = 153 \implies q = 17$.

3. Reflection-Guided Resampling (RGR)

Corrects Neg A using the repair cue.

... $8q + 17 = 10 + q$...

[Correction Trigger]: Wait, *perpen. bisector* means $\angle WPH = 90^\circ$. I cannot equate the acute angles. They must sum to 90° .

Calculation: $(8q + 17) + (10 + q) = 90 \implies 9q + 27 = 90 \implies 9q = 63 \implies q = 7$.

Answer: 7

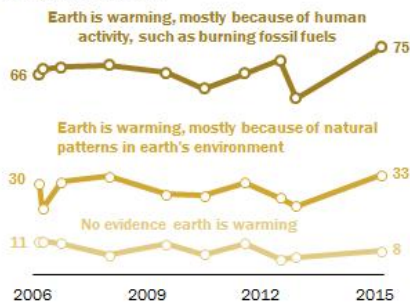
Figure 13. Analysis of Case 1. The Anchor correctly uses the right-angle property. Hard negatives hallucinate isosceles properties or misapply the sum theorem. RGR successfully recovers from the "equality" hallucination.

Case 2: Multi-Point Data Extraction

Concern Up Among Those Who Say Global Warming Due to Human Activity

% saying global warming is a 'very serious' problem

Among those who say...



Survey conducted May 5-June 7, 2015.

PEW RESEARCH CENTER

Question: What's the product of all the values in 2015?

GT Answer: 19800

1. Anchor (y^+) [Positive]

<think> Locating 2015. Three lines visible.

1. Top (Human activity): 75.
2. Middle (Natural patterns): 33.
3. Bottom (No evidence): 8.

Product: $75 \times 33 \times 8 = 2475 \times 8 = 19800$. </think>

<answer> 19800 </answer>

2. Hard Negatives (y^-)

Correct formula application but fails visual grounding.

Neg A (Visual Omission): <think> Reading 2015. Top is 75. Middle is 33.

No other lines clearly visible. Product: $75 \times 33 = 2475$. </think> ✗

Neg B (Data Hallucination): <think> Values are 75, 33, and the bottom looks like 10. Product: $75 \times 33 \times 10 = 24750$. </think> ✗

3. Reflection-Guided Resampling (RGR)

Corrects Neg A (Omission).

<think> ... Product is 2475 ...

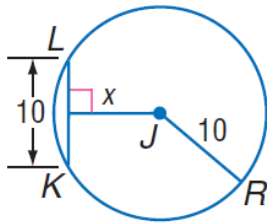
[Correction Trigger]: Re-checking the chart. I missed the bottom beige line labeled 'No evidence'. The endpoint at 2015 is explicitly 8.

Revised values: 75, 33, 8. Calculation: $2475 \times 8 = 19800$. </think>

<answer> 19800 </answer> ✓

Figure 14. Analysis of Case 2. Hard negatives often miss the bottom trend line or misread the axis. RGR triggers a visual re-verification step ("Re-checking the chart") to recover the missing data point.

Case 3: Geometry (Circle Properties)



Question: Find x .

GT Answer: $5\sqrt{3}$ (or approx 8.66)

1. Anchor (y^+) [Positive]

Correctly identifies radius and applies Pythagorean theorem.

<think> The radius is $JR = 10$. The segment from center J to any point on the circle is 10, so hypotenuse to L is 10. The perpendicular segment x bisects chord LK . Total chord is 10, so leg is 5.

Pythagoras: $x^2 + 5^2 = 10^2 \implies x^2 + 25 = 100 \implies x^2 = 75$.

$x = \sqrt{75} = 5\sqrt{3}$. </think>

<answer> $5\sqrt{3}$ </answer>

2. Hard Negatives (y^-)

Valid steps (identifying the triangle) but fails on geometric rules.

Neg A (Equilateral Hallucination): *<think> Chord is 10, Radius is 10. The triangle formed by the chord and center is equilateral. The line x splits the side. Therefore x is simply half of the side length. $x = 10/2 = 5$. </think> ✗*

Neg B (Hypotenuse Confusion): *<think> We have a right triangle. The legs are the distance x and the chord length 10. The radius 10 is the hypotenuse. Wait, actually, we form a triangle with sides 10 and 5. $x = \sqrt{10^2 + 5^2} = \sqrt{125} = 5\sqrt{5}$. </think> ✗*

3. Reflection-Guided Resampling (RGR)

Corrects Neg A (Equilateral assumption).

<think> ... Triangle is equilateral, so x is half the side, which is 5. ...

[Correction Trigger]: *Check the altitude calculation. In an equilateral triangle, the altitude is not half the side; it relates via $\sqrt{3}$.*

Re-calculating: This is a right triangle with hypotenuse 10 (radius) and base 5 (half-chord).

$x^2 = 100 - 25 = 75$. $x = \sqrt{75} = 5\sqrt{3}$. </think>

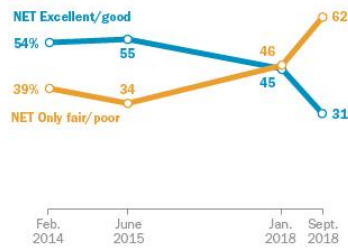
<answer> $5\sqrt{3}$ </answer> ✓

Figure 15. Analysis of Case 3. The model often mistakenly assumes x is a median of length 5 (Neg A) or adds squares instead of subtracting (Neg B). RGR forces a check on the specific right-triangle relationship.

Case 4: Chart Legend Association

U.S. Catholics increasingly critical of the way Pope Francis has handled sex abuse scandal

% of U.S. Catholics who say Pope Francis is doing an _____ job addressing the sex abuse scandal in the Catholic Church



Source: Survey conducted Sept. 18-24, 2018.
"Confidence in Pope Francis Down Sharply in U.S."
PEW RESEARCH CENTER

Question: What's the blue graph represent?

GT Answer: NET Excellent/good

1. Anchor (y^+) [Positive]

<think> I need to identify the label for the blue line.

1. The blue line starts at 54% and ends at 31%.
2. The text colored in blue/teal directly above the start of the line says "NET Excellent/good".
3. The question asks for the representation.

Result: It represents those saying Pope Francis is doing an excellent or good job.

</think>

<answer> NET Excellent/good </answer>

2. Hard Negatives (y^-)

Correct visual attention (looking at lines) but semantic mismatch.

Neg A (Trend Description): <think> The blue line goes down significantly from 54% to 31%. The title says Catholics are "increasingly critical". Therefore, the blue line represents the "Critical" view or the "Sex abuse scandal". </think> ✗

Neg B (Label Swap): <think> There are two lines. The blue one is the lower one at the end (2018). The label nearest to the end of the graph is "NET Only fair/poor" (orange). I will associate the blue line with fair/poor performance. </think> ✗

3. Reflection-Guided Resampling (RGR)

Corrects Neg A (Trend vs Label).

<think> ... The blue line shows the critical trend mentioned in the title. ...

[Correction Trigger]: Distinguish between the chart title and the specific legend entry. Look for text matching the line's color.

Correction: The title describes the trend, but the legend defines the data. The text "NET Excellent/good" is colored teal/blue, matching the line. The line represents positive sentiment. </think>

<answer> NET Excellent/good </answer> ✓

Figure 16. Analysis of Case 4. Hard negatives frequently conflate the chart's narrative title with the specific data series (Neg A) or swap legends due to proximity (Neg B). RGR steers the model to verify color-text alignment.