WHERE AND WHEN: STRUCTURE-GROUNDED CREDIT ASSIGNMENT FOR CROSS-MODAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for visual code generation is cross-modal: the reward comes from rendered pixels, but the actions are code tokens. Most existing methods assign the same advantage to every token in a response and the same credit to every edit in an episode. Finer-grained token-level methods ignore which screen region a token affects. Yet each rendered node, such as a DOM element, links a screen region to a code span. We propose Structure-Grounded Credit Assignment (SGCA), which uses these nodes to locate where in a response and when in an episode the reward is earned. Spatial credit reweights the advantage of each node's tokens according to that node's rendering error: below-average responses are penalized most where they differ from the target, and above-average ones are reinforced most where they match it. Temporal credit scores each edit by its subsequent return minus a learned baseline for edits to that node. On Design2Code, the spatial credit outperforms all baselines, improving CLIP score by 1.53 points, and its gain grows fivefold from the simplest to the most complex pages. In multi-turn repair, the gains from the two credits stack: together they improve CLIP score by 1.37 points and reduce LPIPS by 18% on web pages, and raise the official ChartMimic score, a metric not used in any reward, by 4.3 points. All twelve pre-specified comparisons remain significant after Holm correction.

1 INTRODUCTION

Reinforcement learning with verifiable rewards is the default recipe for post-training reasoning and code models (Shao et al., 2024; Yu et al., 2026). In visual code generation, a vision-language policy writes HTML, SVG, or plotting code, and the reward compares the render with a target image (Si et al., 2025; Li et al., 2026b; Zheng et al., 2026). This setting poses a problem absent from text-only RL, which we call cross-modal credit assignment: the reward lives in pixels, while the actions are code tokens. Although the rendering error is spatially localized, group-relative policy optimization (GRPO) (Shao et al., 2024) collapses it into a single advantage shared by all tokens. As a result, when a response renders the header correctly but breaks the footer, the header tokens are penalized as heavily as the footer tokens (Figure 1). The same collapse recurs across turns. In multi-turn repair, a policy revises its draft turn by turn, and edits are coupled through layout: resizing a container moves its children, while restyling a leaf element changes only itself. Yet existing methods credit each turn by its immediate gain (Jain et al., 2025; Lu et al., 2026b) or by turn-level critics and returns (Zhou et al., 2024; Ekbote et al., 2025), none of which is conditioned on the edited component.

Finer-grained credit exists but is not grounded in the render: it relies on extra rollouts or process rewards, ties tokens to the input image, or needs the action to share the reward's pixel grid (Section 2).

Our key observation is that the render required by the reward already carries the missing correspondence: each DOM node or Matplotlib artist occupies a screen region and comes from a span of source code. Rendered structure is thus a ready-made dictionary between pixels and tokens. We propose Structure-Grounded Credit Assignment (SGCA), the first method to use this dictionary as the credit interface, in both space and time. Spatial credit projects the error of each node through its span onto tokens and redistributes the GRPO advantage within a response: tokens behind deviating regions receive a smaller share of a positive advantage and a larger share of a negative one. Tempo-

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

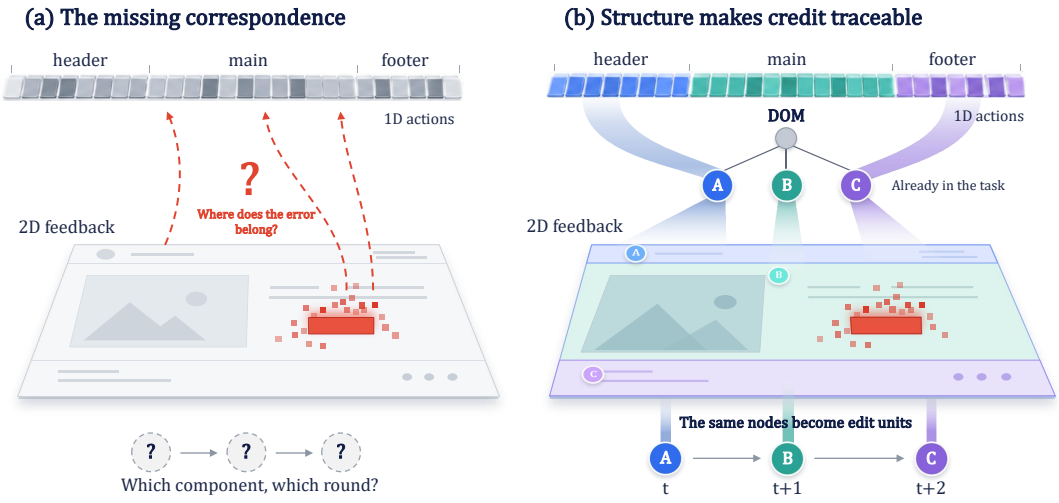

Figure 1: Cross-modal credit assignment in visual code RL. A pixel-space reward reaches the code tokens as one scalar, leaving open which tokens caused an error and which edit earned the return. Rendered structure closes both gaps: each DOM node links a screen region to its source span, and the same nodes serve as the units of editing across rounds.

ral credit measures each node edit by the whole-page return that follows it, with a learned value of editing that node as the baseline; this value also guides which node to edit next. Our contributions are summarized as follows:

- **A structure interface for cross-modal credit assignment.** Rendered nodes map to code tokens through their source spans. The mapping comes free with the render and transfers from web pages to charts.
- **Sign-aware spatial credit.** It redistributes the advantage within a response according to regional error without changing the mean advantage. It needs neither learned parameters nor extra rollouts.
- **Node-conditioned temporal credit for multi-turn repair.** A learned value of editing each node is the baseline for the return that follows. Ranking nodes by this value picks better next edits than ranking by immediate error.

2 Related Work

Visual code generation with executable feedback. Screenshot-to-code has grown from pix2code (Beltramelli, 2018) to large corpora (Laurençon et al., 2024; Gui et al., 2025; Yun et al., 2024) and to benchmarks over pages, vector graphics, and charts (Si et al., 2025; Lu et al., 2026c; Lai et al., 2025; Chen et al., 2025; Rodriguez et al., 2026b; Yang et al., 2025a; Zhang et al., 2025). RL closes the loop through the renderer with multimodal critics (Li et al., 2026b), rankings of rendered candidates (Yang et al., 2025b), patch-level similarity (Zhao et al., 2025a), SVG rendering fidelity (Rodriguez et al., 2026a), or element-wise attribute rewards (Zheng et al., 2026), yet the reward still reaches the policy as one scalar per response. Even methods that decompose the page generate (Wan et al., 2024) or score (Zheng et al., 2026) its parts without crediting them. SGCA redistributes that scalar over tokens, whatever the reward.

Fine-grained credit assignment. GRPO and DAPO broadcast one advantage per response (Shao et al., 2024; Yu et al., 2026). Finer credit comes from extra rollouts (Kazemnejad et al., 2024; Guo et al., 2026), each requiring a render here, or from process rewards (Cui et al., 2025; Cheng et al., 2026), Shapley attribution (Cao et al., 2025), and rubric items (Zhang et al., 2026; Jin et al., 2026), none of which encodes screen position. Execution-grounded credit comes closest but needs a reference program and an LLM localizer (Kumar et al., 2026), and zeroes all other tokens. Multimodal reweighting (Jiao et al., 2026; Li et al., 2026a; Lu et al., 2026a) ties tokens to the input image, not to what they render, and ignores the advantage's sign. Region-level credit in image generation (Yang et al., 2026; Shen et al., 2026) needs action and reward on one latent grid, which code tokens lack. SGCA reads the pixel-to-token map from the render and orients

it by the advantage's sign, without reference code, extra rollouts, or learned attribution.

Multi-turn credit. Multi-turn methods credit whole turns. ArCHer's critic scores whole utterances (Zhou et al., 2024), MURPHY and WebGen-Agent assign retrospective and immediate turn rewards (Ekbote et al., 2025; Lu et al., 2026b), and GiGPO needs states to recur across trajectories (Feng et al., 2026). μCode assumes one-step recoverability (Jain et al., 2025), which local, layout-coupled edits violate. None conditions its baseline on the edited component; SGCA does, with a value computed before sampling, and splits the turn advantage over the edit's tokens.

3 Problem Setup

We study two settings, single-turn generation and multi-turn repair, which share one structural representation of the render.

Structural records. Each rendered node yields one record,

$$\mathcal{Z}_t = \{z_{t,k}\}_{k=1}^{K_t}, \qquad z_{t,k} = \big(\mathrm{tag}_k,\ \mathcal{B}_k,\ [c_k^s, c_k^e),\ d_k\big), \tag{1}$$

where the parser gives the tag and source span, the renderer gives the box $\mathcal{B}_k$, and comparing render and target inside $\mathcal{B}_k$ gives the discrepancy d_k; t indexes rounds in repair and is dropped otherwise. A record thus links a screen region to a code span and needs no reference source. Blocks are the units of spatial credit and additionally carry a normalized error weight u_k (Section 4.1); components are the units edited in repair.

Single-turn generation. A policy π_θ maps a target screenshot x to code $y = (y_1, \ldots, y_L)$, rendered as $I = g(y)$ and scored by a reward $R(x, I, y)$ that combines SigLIP (Zhai et al., 2023), SSIM (Wang et al., 2004), pixel error, and a penalty on unclosed tags, which browsers silently repair (Table A2). GRPO gives every token of response j the same advantage $A_j = (R_j - \mathrm{mean}(R_{1:n}))/(\mathrm{std}(R_{1:n}) + \varepsilon)$ over n samples. SGCA replaces it with a token-level advantage $\widehat{A}_{j,i}$ and leaves the rest of the objective $\mathcal{L}_{\mathrm{S}}$, a clipped surrogate with a KL penalty to π_{ref}, unchanged.

Multi-turn repair. The policy revises a draft over at most T turns, which we call rounds. In round t, a selector picks a component $b_t = h(s_t)$ from the state $s_t = (x, I_t, y_t, \mathcal{Z}_t, T - t)$, and the policy either returns STOP or writes code e_t that replaces the span of b_t. An edit that is malformed or touches code outside b_t is rolled back but still consumes its round. The reward of each round is its gain in similarity to the target. Multi-turn GRPO gives every round, and every token in it, the same normalized episode return. Within an edit, the blocks are the children of b_t, so credit within and across edits operates on adjacent levels of one hierarchy.

4 Method

4.1 Spatial credit: where the error lies in the code

Spatial credit redistributes the group-relative advantage A_j across the tokens of a response. The tokenizer's offset mapping links each block's source span to tokens, so the chain from render to token needs no learned model and reuses the render already computed for the reward.

Blocks are the visible children of the shallowest DOM node with at least two visible children. The target and response renders are compared in luminance to obtain a blurred difference map D. The discrepancy d_k of block k is the mean of D over its box, and its error weight is $u_k = d_k / \max_{k'} d_{k'} \in [0, 1]$. Token i of response j inherits $w_{j,i} = u_k$ from the block k whose span contains it, so $w_{j,i}$ is high inside deviating blocks and low elsewhere (Figure 2C). The weight is then oriented by the sign of the advantage. A response below the group mean is pushed down hardest where it deviates, and one above it is pushed up hardest where it matches,

$$\tilde{w}_{j,i} = \begin{cases} w_{j,i}, & A_j < 0, \\ 1 - w_{j,i}, & A_j \geq 0, \end{cases} \qquad \psi_{j,i} = \eta + (1 - \eta)\,\frac{1 + \alpha\,\tilde{w}_{j,i}}{1 + \alpha\,\overline{\tilde{w}}_j}, \qquad \widehat{A}_{j,i} = A_j\,\psi_{j,i}, \tag{2}$$

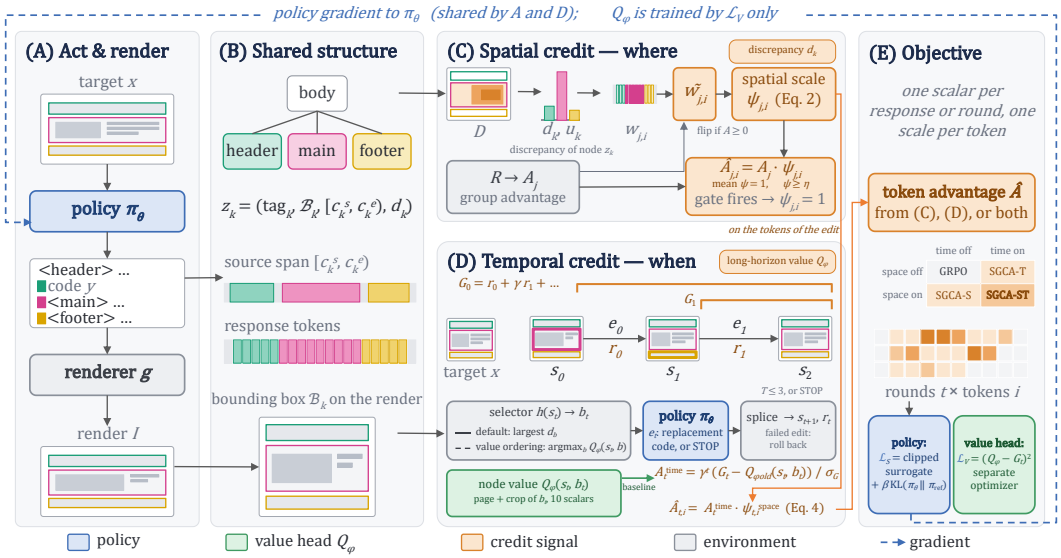

Figure 2: Overview of SGCA. A record z_k links a rendered region to a source span (B). Spatial credit turns node error into a token scale oriented by the sign of the advantage (C). Temporal credit gives each edit its return minus a node value (D), and the two multiply into one token advantage (E).

where $\overline{\tilde{w}}_j$ and $\overline{w}_j$ are the means of $\tilde{w}_{j,i}$ and $w_{j,i}$ over the response tokens and $0 < \eta < 1$. The scale averages to one over these tokens, so the mean advantage of a response equals its GRPO advantage. It is also bounded below by η, so no token is silenced. Rewriting Equation 2 as $\hat{A}_{j,i} = A_j - |A_j| \, \kappa_j \, (w_{j,i} - \overline{w}_j)$ with $\kappa_j \geq 0$ shows what the orientation does. Whatever the sign of A_j, a token whose block deviates more than the response average receives less advantage than one whose block deviates less. The difference grows with $|A_j|$. The strength α sets how much advantage moves between blocks, and $\alpha = 0$ recovers GRPO exactly. We use $\alpha = 3$ and $\eta = 0.25$ throughout.

A gate keeps the GRPO advantage whenever attribution is uninformative: it fires when fewer than two blocks are paired with rendered nodes, when the largest discrepancy is below 0.02, or when the spread of discrepancies is below a quarter of the largest.

Errors can propagate, as when a tall header displaces every block below it. The header then deviates in its own box as well, so its code still receives a large share of the weight. The fault-injection test of Section 5.4 confirms this: the faulty block ranks first on two thirds of pages, despite such displacement. Appendix C.1 gives further details.

4.2 Temporal credit: which edit earned the return

Multi-turn GRPO credits an edit even with rewards earned before it, against a baseline that sees neither the edited component nor the remaining rounds. Temporal credit instead gives each edit the return from its round onward, against a value of the component b_t it edits.

The default selector $b_t = h(s_t)$ takes the editable component with the largest discrepancy d_b in all methods, which thus differ only in credit. The policy sees the target and current renders, the box and source of b_t, and the remaining rounds. The returned code replaces the source span of b_t, and the page is re-rendered and scored by

$$r_t = \Delta\text{SigLIP}_t - \lambda_{\text{mse}} \max\big(\Delta\text{MSE}_t - \tau, \, 0\big), \tag{3}$$

where Δ denotes the change from the previous round, so pixel error is penalized only when it rises by more than τ. Both rewards use SigLIP, so the CLIP evaluation metric is never optimized directly (Appendix A). STOP receives zero and ends the episode, and a failed edit is rolled back with a fixed training penalty.

The node value $Q_\varphi(s_t, b)$ estimates the discounted return $G_t = \sum_{t' \geq t} \gamma^{t'-t} r_{t'}$ when b is edited in round t and the policy acts afterwards. G_t counts gains anywhere on the page, so a layout edit that enables a later text fix shares the gain of that fix. Its inputs are frozen SigLIP embeddings of the two

renders, of their crops to the box of b, and of the render–target differences, plus ten scalars such as d_b and the remaining rounds. Since it uses nothing produced after the selection, it is a valid baseline for the tokens of the edit. The head minimizes $\mathcal{L}_V = \mathbb{E}[(Q_\varphi(s_t, b_t) - G_t)^2]$, first on trajectories of the frozen initial policy and then online (Appendix C.2).

Value and scale then compose into one advantage: the temporal factor sets how much a round is worth, and the spatial scale sets where inside the round that credit goes,

$$A_t^{\text{time}} = \text{sg}\left[\gamma^t \frac{G_t - Q_{\varphi_{\text{old}}}(s_t, b_t)}{\sigma_G}\right], \qquad \boxed{\widehat{A}_{t,i} = A_t^{\text{time}}\ \psi_{t,i}^{\text{space}}} \tag{4}$$

where sg is the stop-gradient, σ_G is the floored standard deviation of the episode returns in the group, φ_{old} is the head before the update, and γ^t discounts every round to the start of the episode. $\psi_{t,i}^{\text{space}}$ is Equation 2 evaluated on the code tokens of round t, with its weight oriented by the sign of A_t^{time}. STOP and end-of-sequence tokens keep scale one. Setting $\psi = 1$ gives SGCA-T; setting A_t^{time} to the normalized episode return gives SGCA-S; doing both gives multi-turn GRPO. The policy is trained by $\mathcal{L}_{\text{S}}$ on $\widehat{A}_{t,i}$, and Q_φ only by $\mathcal{L}_V$ (Algorithm 1).

Value ordering. The same head can also choose the component, $b_t = \arg\max_b Q_\varphi(s_t, b)$. A component is then preferred when its repair unlocks later gains, even if its current error is not the largest. During training the selector takes the maximum with probability 0.8 and a uniformly random component otherwise, so the regression also sees components that the greedy rule would skip. Only executed components receive labels. We evaluate this variant separately (Section 5.4).

4.3 Other structures and cost

Neither credit is specific to HTML. A new domain supplies a task interface: `ParseStructure` returns components with source spans, `LocateRegion` returns the screen region of each, and `ApplyEdit` applies an edit to one component and checks its legality. Together they provide the records of Equation 1 and the repair edits. For charts, a component is the smallest Python statement that creates a visible Matplotlib artist, plus the setter calls that later modify it (Appendix C.3). As for cost, spatial credit needs no extra model pass or render. Temporal credit adds an MLP head with two hidden layers on frozen features, and value ordering adds one head evaluation per component at inference (Table A6).

5 Experiments

5.1 Setup

Training. The base policy is Qwen2.5-VL-7B-Instruct (Bai et al., 2025). Single-turn training runs 3,125 steps (one epoch) on 50,000 WebCode2M pages (Gui et al., 2025), with 16 prompts per step, $n = 4$, and a 2,048-token response limit. Repair training starts from the seed-1 SGCA-S checkpoint, which every condition uses as initial policy and KL reference, and covers 400 WebCode2M pages in one pass. Each page has one fixed first draft, written greedily by the initial policy and shared across conditions and seeds (Appendix D). Its group of four independent trajectories of at most $T = 3$ rounds is followed by one policy update. Conditions that use the node value share one critic warm-up per seed and run four critic updates per group. Every trained condition uses three seeds (Appendix C.4).

Evaluation. Single-turn generation is evaluated on the 484 Design2Code pages (Si et al., 2025) with the final checkpoint and greedy decoding of up to 3,072 tokens. Repair is evaluated on 240 held-out WebCode2M confirmation pages, with one greedy response per round. Repair checkpoints are selected on 240 development pages by cumulative net reward, and confirmation pages are used only for the final evaluation. CLIP uses the OpenAI ViT-L/14 encoder (Radford et al., 2021), and neither CLIP nor LPIPS (Zhang et al., 2018) enters any reward (Appendix A). Quality metrics are averaged over pages rendered validly by all compared methods, and failure rates over all requested pages. Tables report mean ± std over three seeds, with quality metrics scaled by 100. Twelve CLIP differences are pre-specified as primary comparisons (Table A5): two on Design2Code, seven in

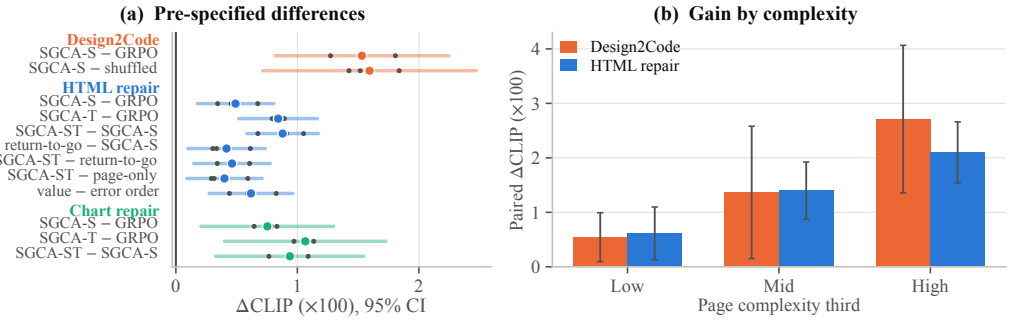

Figure 3: (a) The twelve pre-specified CLIP differences (dots: seeds). (b) Paired CLIP gain by thirds of page complexity. Lines and bars: 95% intervals.

Table 1: Single-turn generation on Design2Code (484 pages). Valid: share of requests that render. †Budget-matched adaptation.

Method	CLIP ↑	SSIM ↑	MSE ↓	LPIPS ↓	HTML ↑	Valid ↑
Qwen2.5-VL-7B-Instruct	70.27	64.50	13.02	32.15	30.69	96.28
GRPO	71.17±0.12	69.99±0.21	10.12±0.20	28.22±0.13	31.91±0.23	96.97
AT-RL† (Jiao et al., 2026)	71.49±0.09	70.28±0.18	10.02±0.13	27.68±0.12	32.18±0.22	96.83
ReLook† (Li et al., 2026b)	71.79±0.07	69.86±0.19	10.08±0.15	27.16±0.19	32.25±0.24	97.04
SGCA-S	**72.70**±0.16	**71.54**±0.23	**8.91**±0.15	**23.78**±0.13	**33.68**±0.21	**98.97**

HTML repair, and three in chart repair. We test them with 50,000 paired bootstrap resamples over seeds and pages and Holm-correct all twelve (Figure 3a).

Baselines. AT-RL (Jiao et al., 2026) and ReLook (Li et al., 2026b) serve as baselines for single-turn generation, and WebGen-Agent (Lu et al., 2026b) and MURPHY (Ekbote et al., 2025) for repair. All four are re-implemented with the same policy, data, renderer, and generation budget, and keep their own feedback and credit rules (Appendix C.4). GRPO, multi-turn GRPO, and the SGCA variants share reward, observation, and action space, so their comparisons isolate the effect of credit.

5.2 Single-turn generation: where the reward belongs

Spatial credit improves on a broadcast advantage. SGCA-S and GRPO differ only in the token scale of Equation 2, so any gap comes from where the advantage lands. SGCA-S improves every metric in Table 1, raising CLIP by 1.53 (Figure 3a); its largest relative gains are on spatial metrics: LPIPS, which no reward uses, falls by 16% and MSE by 12%. Regional credit thus repairs layout rather than exploiting the reward. The HTML tag score also rises, and rendering failures drop by two thirds.

The baselines show that neither reweighting tokens nor refining the reward suffices alone. AT-RL reweights tokens by cross-modal attention, which ignores where the render is wrong, and achieves only a fifth of the gain of SGCA-S. ReLook refines the reward but still broadcasts it, gaining 0.62 CLIP at 2.1 times GRPO's training cost. SGCA-S outperforms both at a lower training cost (Tables A5, A6). What matters is placement: a block-shuffled control, which puts the same weights on wrong spans, gains nothing over GRPO (Section 5.4).

The benefit is largest where a broadcast advantage is weakest. The CLIP gain on the third of pages with the most DOM nodes is five times that on the third with the fewest, and the LPIPS reduction more than triples (Figure 3b, Table A7). The more blocks a page has, the more tokens a single page-level advantage credits wrongly.

Figure 4 shows that the weight tracks visible errors. On page 4518 the footer deviates most and receives the largest weight, and the header, whose hero section is too short, comes second. On page 10848 the weight moves to the main container, which misses most of the question list. Blame thus falls on the faulty region's code, and credit on the rest.

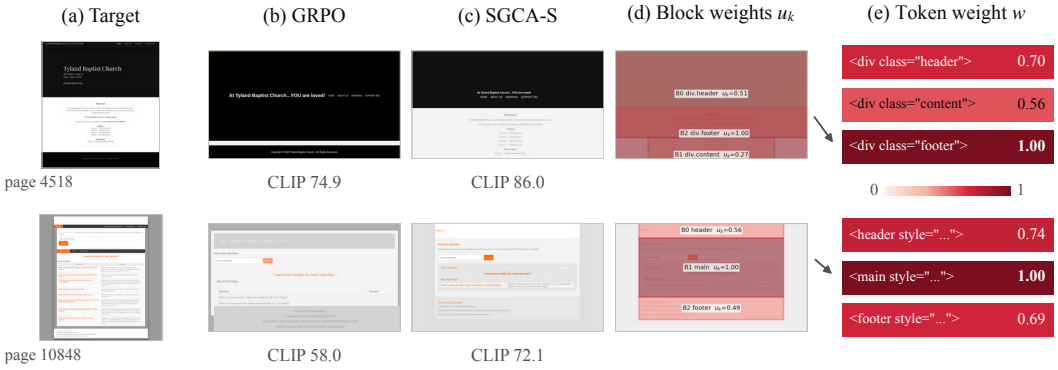

Figure 4: Spatial credit on two Design2Code pages. (d) Error weight of each rendered block. (e) Error weight carried onto the source span of each block. Arrow: block with the largest weight.

Table 2: HTML repair with free-form code edits (240 pages, three rounds, shared drafts). Reward: cumulative net reward. Fail.: trajectories with a failed edit (%). †Budget-matched adaptation.

Method	Space	Time	CLIP ↑	MSE ↓	LPIPS ↓	Reward ↑	Rounds	Fail. ↓
No-edit (shared draft)	–	–	83.19	4.15	18.88	0	0	–
WebGen-Agent† (Lu et al., 2026b)	–	–	84.18±0.06	3.98±0.07	17.67±0.18	0.70	2.42	2.1
MURPHY† (Ekbote et al., 2025)	–	–	84.52±0.05	3.94±0.05	16.29±0.14	1.00	2.36	1.5
Multi-turn GRPO	○	○	83.93±0.07	4.01±0.04	17.96±0.22	0.62	2.47	2.5
SGCA-S	●	○	84.42±0.13	3.98±0.06	17.11±0.43	0.89	2.36	1.8
SGCA-T	○	●	84.77±0.03	3.99±0.05	15.94±0.35	1.18	2.33	1.7
SGCA-ST	●	●	**85.30**±0.07	**3.79**±0.08	**14.80**±0.16	**1.61**	2.26	**1.0**

5.3 Multi-turn repair: which edit earned it

Both credits improve repair, and their gains add. The four internal conditions of Table 2 differ only in the two switches of Equation 4, and all three primary differences survive Holm correction (Figure 3a). Each credit adds almost the same CLIP whether the other is off or on: 0.49 and 0.53 for spatial credit, 0.84 and 0.88 for temporal credit. The two act on the same tokens but answer different questions: which edit earned the return, and which tokens inside that edit did. Temporal credit is the larger factor, as the selector already confines each edit to one component. SGCA-ST is best on every quality metric, with the lowest failure rate, 1.0% against 2.5% for multi-turn GRPO. It also makes fewer edits, so its reward per edit is nearly three times that of multi-turn GRPO.

Node-level credit outperforms turn-level credit. WebGen-Agent and MURPHY credit whole rounds. WebGen-Agent uses an immediate critic score, which misses gains an edit enables later; MURPHY uses the best leaf return of a rollout tree, which credits a round with its best-case continuation. Neither accounts for which node an edit changes. Despite about twice the training time of GRPO (Table A6), they finish 1.12 and 0.77 CLIP below SGCA-ST, with intervals that exclude zero (Table A5).

The gap widens over rounds and with page complexity. Temporal credit helps from the first round, doubling the first-round gain from 0.47 CLIP under multi-turn GRPO to 0.94 (Figure 5a), since a first edit is now credited with the fixes it enables. The gap between SGCA-ST and multi-turn GRPO then nearly doubles over the next two rounds, to 1.37 CLIP, as a late edit no longer inherits the gains of earlier edits. The gap also grows from 0.61 CLIP on the simplest third of pages to 2.10 on the most complex (Figure 3b), where each edit changes a smaller share of the page, so a page-level return reveals least about any single edit.

The gain survives compute matching. Extra compute does not replace credit. Trained to the full cost of SGCA-ST on seed 1, multi-turn GRPO closes under half of the CLIP gap and a sixth of the LPIPS gap, and still trails by 0.71 CLIP with an interval that excludes zero (Appendix F). SGCA-ST needs 1.78 times the training GPU time of multi-turn GRPO and only 1.09 times its inference time.

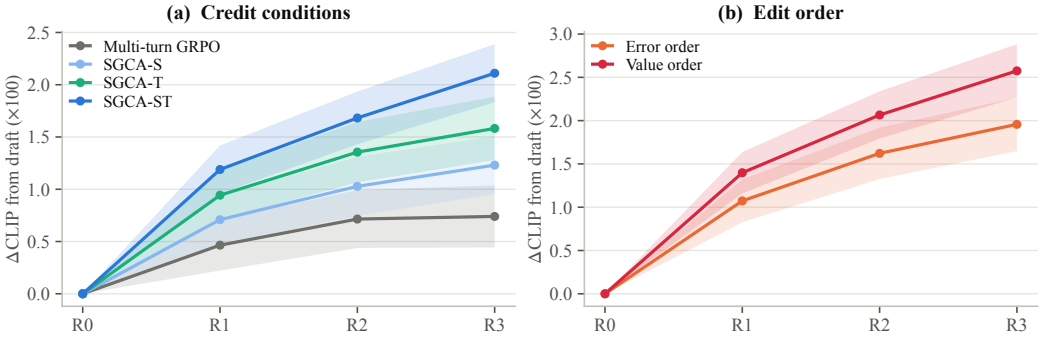

Figure 5: CLIP gain over the draft by round in HTML repair. (a) Credit conditions. (b) Value and error order from the same checkpoints. Bands: 95% intervals.

Table 3: Ablations. Top: orientation and placement of spatial credit (Design2Code). Bottom: return, value input, and edit order (HTML repair, spatial credit on). Err.: largest current error. Fail.: as in Table 2. Last row: checkpoints of the row above with the other selector.

Token weights (single-turn, 484 pages)		CLIP ↑	SSIM ↑	MSE ↓	LPIPS ↓	HTML ↑
Uniform (GRPO)		71.17±0.12	69.99±0.21	10.12±0.20	28.22±0.13	31.91±0.23
Sign-agnostic weight		71.56±0.08	70.83±0.15	9.77±0.21	25.36±0.15	32.29±0.27
Block-shuffled		71.11±0.06	69.81±0.30	10.31±0.27	28.67±0.17	28.25±0.31
Sign-aware weight, correct map (SGCA-S)		**72.70**±0.16	**71.54**±0.23	**8.91**±0.15	**23.78**±0.13	**33.68**±0.21
Temporal signal (repair, 240 pages)	**Selector, train / test**	**CLIP ↑**	**MSE ↓**	**LPIPS ↓**	**Reward ↑**	**Fail. ↓**
Shared episode return (SGCA-S)	Err. / Err.	84.42±0.13	3.98±0.06	17.11±0.43	0.89	1.8
Return-to-go, no critic	Err. / Err.	84.83±0.15	3.90±0.03	15.89±0.18	1.13	1.8
Page-only critic	Err. / Err.	84.90±0.10	3.91±0.03	16.03±0.18	1.21	1.5
Node critic (SGCA-ST)	Err. / Err.	85.30±0.07	3.79±0.08	14.80±0.16	1.61	1.0
Node critic, value ordering	Q_φ / Q_φ	**85.76**±0.09	**3.69**±0.05	**13.93**±0.10	**2.06**	**0.7**
same checkpoints	Q_φ / Err.	85.14±0.10	3.93±0.07	15.79±0.16	1.32	1.4

5.4 Ablations: placement, value, and order

The gain of spatial credit comes from its placement and its orientation. Two controls each remove one property of the weights (Table 3, top). The block-shuffled control permutes the block weights within each response, breaking only the block-to-token match. It ends 0.06 CLIP below GRPO, so the gain cannot be a side effect of a non-uniform scale. Misplaced weights even harm structure: the HTML tag score drops 3.7 points below GRPO, because credit lands on code that did not cause the error. The sign-agnostic weight keeps the placement but applies $w_{j,i}$ regardless of the sign of A_j, so a better-than-average response is reinforced most where it deviates from the target (Appendix C.1). It retains a quarter of the gain, 0.39 of 1.53 CLIP. Orientation is what makes good responses reinforced where they are right and poor ones penalized where they are wrong. A fault-injection test on 100 pages, with 25 candidate blocks on average, checks localization directly. The token weight ranks the faulty block first on 67% of pages, against 1% for shuffled weights and 4%, the chance level, for uniform ones (Appendix F).

The temporal gain needs the return that follows an edit and a value that sees the node. Temporal credit makes two changes: each round receives the return that follows it, and that return is measured against a node value. Two ablations separate them (Table 3, bottom). Return-to-go without a critic centers the return of each round over its group and gains 0.42 CLIP over SGCA-S, because a late edit no longer inherits earlier gains and its own contribution becomes visible. The node critic adds a further 0.46. A page-only critic with the same capacity and training budget adds only 0.07 and stays 0.40 below the node critic, so the gain comes from the node input, not from having a critic. A value of the whole page cannot predict what an edit to one component should earn; a value of the edited node can. All three differences survive Holm correction (Table A5).

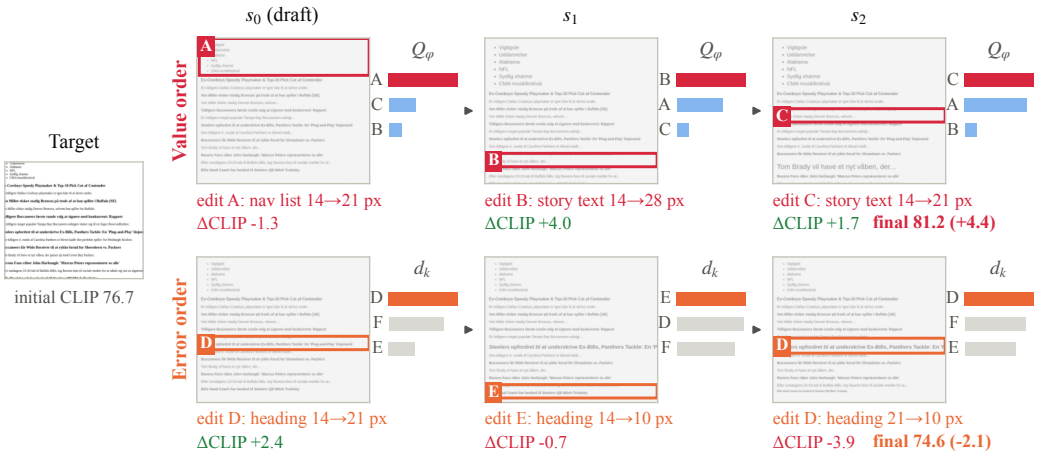

Figure 6: Edits chosen by value (top) and by current error (bottom) from one checkpoint on the same page. Box: edited component. Bars: the ranking that selects it.

Table 4: Chart repair on ChartMimic (600 charts, three rounds, shared drafts). Low-level: official ChartMimic score, outside the reward. Reward, Rounds, and Fail.: as in Table 2.

Method	Space	Time	Low-level ↑	CLIP ↑	MSE ↓	LPIPS ↓	Reward ↑	Rounds	Fail. ↓
No-edit (shared draft)	–	–	72.52	74.07	6.42	28.87	0	0	–
Multi-turn GRPO	○	○	74.59±0.09	76.02±0.06	5.71±0.08	26.08±0.21	1.48	2.52	2.8
SGCA-S	●	○	76.35±0.52	76.77±0.11	5.35±0.08	24.21±0.21	2.28	2.44	2.2
SGCA-T	○	●	77.12±0.25	77.09±0.04	5.15±0.07	23.13±0.18	2.59	2.49	2.0
SGCA-ST	●	●	**78.92**±0.46	**77.71**±0.10	**4.89**±0.05	**21.88**±0.15	**3.18**	2.33	**1.4**

The node value orders edits better than the immediate error does. To separate the order from the learned policy, the value-ordering checkpoints are also evaluated with the largest-error selector. Value ordering leads after every round (Figure 5b) and ends 0.62 CLIP and 0.74 cumulative reward higher, a difference that survives Holm correction, with half the failure rate. With the policy fixed, the gain comes from the order alone: the value anticipates which edits enable later ones. In Figure 6, value ordering accepts an early loss and ends 4.4 CLIP above the draft, while error ordering ends 2.1 below it (Appendix F). Trained as its own condition, value ordering is the best overall, 0.47 CLIP above SGCA-ST (Table 3).

5.5 Beyond HTML: chart code

The gains carry over to chart code on ChartMimic (Yang et al., 2025a). Only the task interface changes: components are Matplotlib artists (Section 4.3). A policy fine-tuned on 1,600 Chart2Code-160k programs (Zhao et al., 2025b) writes shared drafts for all 600 test charts; other settings follow HTML repair (Appendix C.4). All three primary differences survive Holm correction: 0.75 CLIP for spatial credit, 1.07 for temporal credit, and 0.94 for adding temporal to spatial credit (Table 4). As on pages, temporal credit dominates and their gains add. Relative to multi-turn GRPO, SGCA-ST halves the failure rate and gains 4.3 points on the low-level score, which no reward uses, because crediting individual artists repairs the text, colors, and layout that this score checks per element.

6 Conclusion

In visual code RL, rewards live in pixels and actions in code tokens, so a scalar reward hides which tokens, or which edit, caused an error. The render resolves this: its structure links each region to the code that drew it. SGCA exploits this link twice: in space, it places blame and credit on each region's code; in time, it credits each edit against its node's value. Both credits pay off on pages and charts, their gains add, and controls attribute each gain to this structure. Other rendered code has the same link, so structure-grounded credit is a general recipe for learning from visual feedback.

AI USE STATEMENT

In this work, generative AI tools were used for editing and polishing the manuscript text and for assistance with LaTeX formatting. They were not used to generate research ideas, experimental designs, or numerical results. All AI-assisted text was reviewed and revised by the authors, and all claims were checked against the experimental logs by at least two authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work trains models to reproduce web pages and charts from images. All training and evaluation data come from public datasets released for research use, and no personal data was collected. Generated code is rendered in a sandboxed browser or an isolated Python process, without network access.

REPRODUCIBILITY STATEMENT

Section 5.1 and Appendix C give the training configuration, the attribution pipeline, the edit protocol, and the adaptation of each baseline. Appendix E reports per-seed results and the statistical protocol, and Appendix D the data construction and splits. Code is available at an anonymous repository (`https://anonymous.4open.science/r/sgca-BD7D`) and will be publicly released upon acceptance.

REFERENCES

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, 2025. URL `https://arxiv.org/abs/2502.13923`.

Tony Beltramelli. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pp. 1–6, 2018.

Meng Cao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Scar: Shapley credit assignment for more efficient rlhf. *arXiv preprint arXiv:2505.20417*, 2025.

Siqi Chen, Xinyu Dong, Haolei Xu, Xingyu Wu, Fei Tang, Hang Zhang, Yuchen Yan, Linjuan Wu, Wenqi Zhang, Guiyang Hou, et al. Svgenius: Benchmarking llms in svg understanding, editing and generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 13289–13296, 2025.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *Advances in Neural Information Processing Systems*, 38:131646–131671, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Chanakya Ekbote, Vijay Lingam, Sujay Sanghavi, Jun Huan, Behrooz Omidvar-Tehrani, Anoop Deoras, and Stefano Soatto. Murphy: Feedback-aware grpo with retrospective credit assignment for multi-turn code generation. *arXiv preprint arXiv:2511.07833*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Bohua Chen, Yi Su, Dongping Chen, Siyuan Wu, Xing Zhou, et al. Webcode2m: A real-world dataset for code generation from webpage designs. In *Proceedings of the ACM on Web Conference 2025*, pp. 1834–1845, 2025.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *Advances in Neural Information Processing Systems*, 38:114399–114431, 2026.

Arnav Kumar Jain, Gonzalo Gonzalez-Pumariega, Wayne Chen, Alexander M Rush, Wenting Zhao, and Sanjiban Choudhury. Multi-turn code generation through single-step rewards. *arXiv preprint arXiv:2502.20380*, 2025.

Zhengbo Jiao, Shaobo Wang, Zifan Zhang, Wei Wang, Bing Zhao, Hu Wei, and Linfeng Zhang. Credit where it is due: Cross-modality connectivity drives precise reinforcement learning for mllm reasoning. *arXiv preprint arXiv:2602.11455*, 2026.

Rui Jin, Jikai Chen, Yihan Chen, Hao Zhou, Demin Zhu, Kaichen Yang, Dong Wang, Linjian Mo, and Chenyi Zhuang. Rubric-to-code credit assignment for reinforcement learning. *arXiv preprint arXiv:2608.27906*, 2026.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. 2024.

Abhijit Kumar, Natalya Kumar, and Shikhar Gupta. Execution-grounded credit assignment for grpo in code generation. *arXiv preprint arXiv:2603.16158*, 2026.

Peichao Lai, Jinhui Zhuang, Kexuan Zhang, Ningchang Xiong, Shengjie Wang, Yanwei Xu, Chong Chen, Yilei Wang, and Bin Cui. Webrenderbench: Enhancing web interface generation through layout-style consistency and reinforcement learning. *arXiv preprint arXiv:2510.04097*, 2025.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into html code with the websight dataset. *arXiv preprint arXiv:2403.09029*, 2024.

Qiming Li, Tianlun Li, Xiaolong Cheng, Hangyu Li, Ruiyan Gong, Kangning Niu, Kaitao Jiang, and Mu Xu. Prpo: Perception-reinforced policy optimization via token-level dynamic advantage reshaping. *arXiv preprint arXiv:2606.08708*, 2026a.

Yuhang Li, Chenchen Zhang, Ruilin Lv, Ao Liu, Ken Deng, Yuanxing Zhang, Jiaheng Liu, and Bo Zhou. Relook: Vision-grounded rl with a multimodal llm critic for agentic web coding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25471–25485, 2026b.

Jinda Lu, Junkang Wu, Jinghan Li, Kexin Huang, Shuo Yang, Guoyin Wang, Jiancan Wu, Xiang Wang, and Xiangnan He. Bridging perception and reasoning: Token reweighting for rlvr in multimodal llms. *arXiv preprint arXiv:2603.25077*, 2026a.

Zimu Lu, Houxing Ren, Yunqiao Yang, Ke Wang, Zhuofan Zong, Junting Pan, Mingjie Zhan, and Hongsheng Li. Webgen-agent: Enhancing interactive website generation with multi-level feedback and step-level reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 83478–83525, 2026b.

Zimu Lu, Yunqiao Yang, Houxing Ren, Haotian Hou, Han Xiao, Ke Wang, Weikang Shi, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Webgen-bench: Evaluating llms on generating interactive and functional websites from scratch. *Advances in Neural Information Processing Systems*, 38, 2026c.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Rishav Pramanik, Aarash Feizi, Pascal Wichmann, Arnab Mondal, Mohammad R Samsami, Rabiul Awal, Perouz Taslakian, et al. Rendering-aware reinforcement learning for vector graphics generation. *Advances in Neural Information Processing Systems*, 38:60496–60534, 2026a.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Tianyang Zhang, Rishav Pramanik, Meng Lin, Xiaoqing Xie, Marco Terral, Darsh Kaushik, Aly Shariff, et al. Vectorgym: A multitask benchmark for svg code generation, sketching, and editing. *arXiv preprint arXiv:2603.29852*, 2026b.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jinjie Shen, Wei Deng, Xian Hu, Daiguo Zhou, and Jian Luan. Star: Spatiotemporal adaptive reward allocation for text-to-image rl post-training. *arXiv preprint arXiv:2606.17979*, 2026.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2code: Benchmarking multimodal code generation for automated front-end engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3956–3974, 2025.

Yuxuan Wan, Chaozheng Wang, Yi Dong, Wenxuan Wang, Shuqing Li, Yintong Huo, and Michael R Lyu. Automatically generating ui code from screenshot: A divide-and-conquer-based approach. *arXiv preprint arXiv:2406.16386*, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, et al. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *International Conference on Learning Representations*, volume 2025, pp. 26590–26646, 2025a.

Yankai Yang, Yancheng Long, Wei Chen, Xingyu Lu, Hongyang Wei, Bin Wen, Fan Yang, Tingting Gao, Han Li, and Shuo Yang. Spatialflow-grpo: Where spatial credit drives image editing. *arXiv preprint arXiv:2606.26872*, 2026.

Zhen Yang, Wenyi Hong, Mingde Xu, Xinyue Fan, Weihan Wang, Jiale Cheng, Xiaotao Gu, and Jie Tang. Ui2codeˆ n: Ui-to-code generation as interactive visual optimization. *arXiv preprint arXiv:2511.08195*, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Sukmin Yun, Haokun Lin, Rusiru Thushara, Mohammad Q Bhat, Yongxin Wang, Zutao Jiang, Mingkai Deng, Jinhong Wang, Tianhua Tao, Junbo Li, et al. Web2code: A large-scale webpage-to-code dataset and evaluation framework for multimodal llms. *Advances in neural information processing systems*, 37:112134–112157, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Bo-Wen Zhang, Junwei He, Wen Wang, Song-Lin Lv, Wentao Ma, Rongyi Lin, Shuhan Zhong, and Lan-Zhe Guo. Cort: Counterfactual replay for token-level rubric-guided policy optimization. *arXiv preprint arXiv:2607.25659*, 2026.

Jiajun Zhang, Jianke Zhang, Zeyu Cui, Jiaxi Yang, Lei Zhang, Binyuan Hui, Qiang Liu, Zilei Wang, Liang Wang, and Junyang Lin. Plotcraft: Pushing the limits of llms for complex and interactive data visualization. *arXiv preprint arXiv:2511.00010*, 2025.

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

Xuanle Zhao, Deyang Jiang, Zhixiong Zeng, Lei Chen, Haibo Qiu, Jing Huang, Yufeng Zhong, Liming Zheng, Yilin Cao, and Lin Ma. Vincicoder: Unifying multimodal code generation via coarse-to-fine visual reinforcement learning. *arXiv preprint arXiv:2511.00391*, 2025a.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7333–7348, 2025b.

Yaozhi Zheng, Yilei Jiang, Manyuan Zhang, Yuxuan Wan, Kaituo Feng, Tianshuo Peng, Bo Zhang, and Xiangyu Yue. Unicoder: Unified visual-to-code generation via symbolic rewards and reference-guided code optimization. *arXiv preprint arXiv:2606.31732*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024.

A METRIC DEFINITIONS

Generation and repair. CLIP is the cosine similarity of the OpenAI CLIP ViT-L/14 image embeddings of the generated and the target screenshot (Radford et al., 2021); both rewards use SigLIP (Zhai et al., 2023) instead, so CLIP enters no reward. SSIM, MSE, and LPIPS (Zhang et al., 2018) are computed at 1280×720. HTML is the sequence similarity of the tag names of the generated and the reference page. Cumulative net reward is the undiscounted sum of the per-round rewards, to which STOP and failed edits contribute zero. Rounds counts attempted edits, and the failure rate is the share of pages whose trajectory contains a failed edit.

Charts and fault localization. The official low-level score is the mean of the text, layout, chart-type, and color F1 of the ChartMimic evaluator (Yang et al., 2025a), multiplied by 100. In the fault test, a block is scored by the mean token weight w over its span, before orientation. Top-1 is the share of pages on which the faulty block scores highest, and MRR is its mean reciprocal rank.

B CLOSEST METHODS

Table A1 places SGCA among the closest methods.

Table A1: Methods closest to SGCA, by source of the fine-grained signal and unit of credit.

Method (domain)	Source of the fine-grained signal	Unit of credit	Relation to SGCA
SpatialFlow-GRPO (image editing)	region-aware reward model	latent regions	same redistribution principle; region and action share coordinates
AT-RL (multimodal reasoning)	cross-modal attention connectivity	token	model-internal token weights; compared in Table 1
ReLook (web code)	multimodal critic	response	finer reward, broadcast advantage; compared in Table 1
ArCHer (LLM agents)	utterance-level off-policy critic	turn	same return-and-baseline framework; the edited node is not among the critic inputs
MURPHY (multi-turn code)	best descendant return in a rollout tree	turn	return propagation over turns; compared in Table 2
μCode (multi-turn code)	single-step verifier	turn	assumes one-step recovery; layout coupling breaks it
SGCA (visual code)	renderer and structure parser	token and edit	sign-aware error weight on tokens, times a return with a node-conditioned baseline

Algorithm 1 Multi-turn repair rollout and credit computation for one trajectory.

Require: target x, draft y_0, budget T, policy π_θ, selector h, head Q_φ, executor

render y_0 to $(I_0, \mathcal{Z}_0)$ and set $s_0 = (x, I_0, y_0, \mathcal{Z}_0, T)$
for $t = 0, \dots, T-1$ **do**
  rebuild the editable components from y_t and $\mathcal{Z}_t$; end the episode if there is none
  $b_t \leftarrow h(s_t)$: the largest d_b, or $\arg\max_b Q_\varphi(s_t, b)$ with exploration under value ordering
  cache $q_t \leftarrow Q_\varphi(s_t, b_t)$ before any token of the edit is sampled
  sample $e_t \sim \pi_\theta(\cdot \mid x, I_t, b_t, \text{source}, T-t)$; if $e_t = \text{STOP}$, set $r_t = 0$ and end the episode
  check the action and splice its code into the source span of b_t; re-render to $(I_{t+1}, \mathcal{Z}_{t+1})$
  on failure roll back to y_t and use the penalty; otherwise compute r_t by Equation 3
end for
compute G_t backward and A_t^{time} from the cached q_t; orient the weights of round t by the sign of A_t^{time} and compute $\widehat{A}_{t,i}$ by Equation 4
after the group: one policy update with $\mathcal{L}_{\text{S}}$, then four updates of Q_φ with $\mathcal{L}_V$ on the same labels

C Implementation details

C.1 Spatial credit

Objective. The policy objective $\mathcal{L}_{\text{S}}$ of Section 3 is the clipped surrogate with a KL penalty to the initialization,

$$\mathcal{L}_{\text{S}} = -\mathbb{E}_{x,j} \frac{1}{L_j} \sum_i m_{j,i} \min\Big(\rho_{j,i} \widehat{A}_{j,i},\ \text{clip}(\rho_{j,i}, 1-\varepsilon_c, 1+\varepsilon_c)\, \widehat{A}_{j,i}\Big) + \beta\, \mathcal{L}_{\text{KL}}(\pi_\theta, \pi_{\text{ref}}), \quad (5)$$

where $\rho_{j,i}$ is the token probability ratio, $m_{j,i}$ the response mask, and L_j the length of response j. In repair, (j, i) runs over the rounds and tokens of the trajectories in a group. Setting $\widehat{A}_{j,i} = A_j$ recovers GRPO. No gradient flows through the renderer or the parser, and the KL term is not scaled.

Attribution pipeline. Pages are rendered in a headless browser at 1280×720 with scripts disabled, and one render serves the reward, the attribution, and the next state. The block level is found by descending through at most three single-child wrappers. Source blocks are paired with rendered DOM nodes in document order, and D is the blurred luminance difference of the two renders. A token takes the weight of the block that contains its first character, and tokens outside every block take the mean block weight. The mean $\overline{\tilde{w}}_j$ of Equation 2 is taken over all response tokens, so the scale averages to one over them.

Edit payloads. In repair the scale is computed on the code of the edit, whose blocks are the visible children of the replaced component after the edit. A response whose mapping is unreliable keeps scale one throughout.

Block-shuffled control. The block weights of each response are randomly permuted before the token alignment. Gate, orientation, strength, and floor are those of SGCA-S.

Sign-agnostic weight. The sign-agnostic row of Table 3 applies $w_{j,i}$ in place of $\tilde{w}_{j,i}$ in Equation 2, so a response above its group is pushed up hardest where it deviates. Strength α, floor η, gate, blocks, and token alignment are those of SGCA-S.

C.2 Temporal credit

Actions and observation. Editable components are the visible children of `body`, split into their children when they exceed 1,024 tokens. The policy returns one complete HTML root element or STOP. The element may change the descendants of the component but nothing outside it, and a violation is a failed edit. Each round is prompted with both screenshots, the remaining rounds, and the box and source of the selected component, without reference source, reward, or value prediction.

Reward, failures, and advantages. A failed edit is rolled back, consumes its round, and is penalized in the training reward only. All conditions normalize by the same group statistics of the dis-

counted episode return. Multi-turn GRPO thus optimizes the same return, and the temporal switch changes only how it is assigned to rounds. The return-to-go ablation gives round t the return that follows it, centered over the trajectories of the group still active in round t.

Node value head and its training. The head is an MLP with two hidden layers on frozen SigLIP embeddings of the target, the current render, and their difference, the same three for the crop to the component, and ten scalars such as the box geometry, d_b, and the remaining rounds. It shares no parameter with the policy. It is warmed up on rollouts of the frozen initial policy, which all conditions and seeds of a domain share, and then updated four times per group on the labels of that group.

Value ordering and the page-only critic. Under value ordering all editable components are scored, and evaluation takes the maximum. The order comparison evaluates the same checkpoint once with each selector. The page-only critic has the architecture of the node head, with the crop embeddings replaced by page embeddings and the component scalars set to zero.

C.3 Chart adapter

The span of a chart component is read from the abstract syntax tree, and its box is the union of its artist boxes after drawing. An edit replaces the code of one component and may change its data, text, colors, and positions, but not imports, other components, or the figure layout. Execution errors, blank figures, and timeouts are failed edits.

C.4 Configuration and baselines

Table A2 lists the configuration and Table A3 the adaptation of each external method. The methods with visual feedback share one frozen vision-language critic, which the internal conditions never see.

D Data

Splits are defined on source pages (Table A4). Repair pages are drawn from disjoint shards of the frozen WebCode2M pool: 400 training pages from train0–6, 240 development pages from confirm16–18, and 240 confirmation pages from confirm19–21. The draft of each page is the greedy output of the SGCA-S checkpoint of seed 1 at step 3,125 on the target screenshot, with at most 2,048 tokens, frozen before any repair training. Chart records must be executable, and near duplicates of test charts are removed before the split into 200 development and 1,600 training programs.

E Statistics and per-seed results

Main tables report the mean and the standard deviation over three training seeds. Differences are paired by page, and intervals come from 50,000 bootstrap resamples of seeds and pages with shared indices for both methods. The twelve pre-specified CLIP differences are Holm-corrected at level 0.05, and all twelve are significant (Table A5); the other differences are descriptive. Table A6 gives every trained run.

F Additional analyses

Page complexity. Per-page differences are split into thirds by the number of visible DOM nodes, counted on the reference page for Design2Code and on the shared draft for repair (Figure 3b and Table A7).

Compute-matched GRPO. Multi-turn GRPO is continued from its final state in HTML repair, seed 1, until its training cost matches that of SGCA-ST including the full critic warm-up of that seed (Table A8).

Table A2: Training configuration.

Item	Single-turn generation	Multi-turn repair (HTML; charts where different)
Initial policy	Qwen2.5-VL-7B-Instruct	SGCA-S checkpoint of seed 1, step 3,125; charts: domain SFT checkpoint
Data	50,000 WebCode2M pages	400 pages with fixed drafts; charts: 400 of 1,600 programs per seed
Budget	3,125 steps, 16 prompts per step, $n = 4$	400 groups, four trajectories per group, $T = 3$
Learning rate, KL, clip	10^{-6}, $\beta = 0.05$, $\varepsilon_c = 0.2$	same
Prompt / response limit	1,536 / 2,048 tokens	12,288 / 2,048 tokens per round; charts: 4,096
Evaluation decoding	greedy, 3,072 tokens, step 3,125	greedy, one response per round, selected checkpoint
Spatial credit	sign-aware weight, $\alpha = 3$, $\eta = 0.25$, wrapper depth 3	same, on payload tokens, oriented by A_t^{time}
Attribution gate	≥ 2 matched blocks, $\max_k d_k \geq 0.02$, $(\max_k d_k - \min_k d_k)/\max_k d_k \geq 0.25$	same
Generation reward	$R = 0.85R_{\text{vis}} + 0.15R_{\text{str}}$, $R_{\text{vis}} = 0.4R_{\text{sem}} + 0.3R_{\text{ssim}} + 0.3R_{\text{pix}}$; sem: SigLIP, ssim: SSIM, pix: pixel-error term, str: tag completeness	–
Repair reward	–	$\lambda_{\text{mse}} = 5$, $\tau = 5 \times 10^{-4}$, STOP 0, failed edit -0.05 in training
Discount, return scale	–	$\gamma = 0.99$, $\sigma_G \geq 0.01$
Node value head	–	(6 SigLIP embeddings + 10) $\to 256 \to 256 \to 1$, AdamW 10^{-4}, four updates per group
Hardware	8 × H100 (80 GB) per run, BF16, FSDP2	same; value head, returns, and losses in FP32
Seeds	three training seeds per condition	three training seeds per condition

Table A3: Budget-matched adaptations of the external methods.

Method	Training	Inference
AT-RL	four responses per prompt under the single-turn reward; token weights from attention debiasing, graph clustering, and soft credit	one response of 3,072 tokens
ReLook	two drafts per prompt with one critic-guided revision each, accepted only with a higher critic score; GRPO on the accepted paths and a feedback loss with weight 0.1	one response of 3,072 tokens, without critic or revision
WebGen-Agent (Step-GRPO)	four trajectories of at most three steps with visual feedback; the critic score of each step, normalized over the group, is its advantage; no KL term	one greedy trajectory; the state with the best critic score is returned
MURPHY (MaRS)	a rollout tree with branching $[2, 2, 1]$ and at most ten generations; the best leaf return is propagated to the ancestors	one greedy trajectory of at most three rounds

Spatial credit on two further pages. Figure A1 repeats Figure 4 on pages 6957 and 9610. On page 6957 the content block, which lacks the product image of the target, receives the largest weight. On page 9610 it is the footer region, where the banner above the footer is missing. On both pages the largest token weight falls on the source span of that block.

Edit order on one page. Figure A2 crops the page of Figure 6 around each edited component and sets the region before and after the edit beside the same region of the target.

Table A4: Datasets and their roles.

Dataset	Stage	Size	Use
WebCode2M fixed subset (Gui et al., 2025)	single-turn RL	50,000 pages	all rows of Table 1, block-shuffled control
WebCode2M held-out pages	development	100 of 500 held-out pages	fault injection
Design2Code (Si et al., 2025)	single-turn evaluation	484 pages	Tables 1 and 3, Figures 4 and A1
WebCode2M repair pages with fixed drafts	repair	400 train, 240 dev, 240 confirmation	Tables 2 and 3, Figures 5 and 3
Chart2Code-160k (Zhao et al., 2025b)	chart SFT and repair	1,600 train, 200 dev	domain start, drafts, repair training
ChartMimic Direct Mimic (Yang et al., 2025a)	chart evaluation	600 charts	Table 4

Table A5: Paired CLIP differences (95% intervals). Top: pre-specified comparisons, Holm-corrected. Bottom: descriptive comparisons. N: valid paired pages.

Comparison	Task	N	ΔCLIP	95% CI	p (Holm)
SGCA-S − GRPO	Design2Code	447	1.53	[0.82, 2.25]	<0.001
SGCA-S − block-shuffled	Design2Code	447	1.59	[0.71, 2.47]	0.004
SGCA-S − multi-turn GRPO	HTML repair	238	0.49	[0.18, 0.81]	0.013
SGCA-T − multi-turn GRPO	HTML repair	238	0.84	[0.52, 1.17]	<0.001
SGCA-ST − SGCA-S	HTML repair	238	0.88	[0.58, 1.17]	<0.001
Return-to-go − SGCA-S	HTML repair	238	0.42	[0.10, 0.74]	0.021
SGCA-ST − return-to-go	HTML repair	238	0.46	[0.15, 0.78]	0.016
SGCA-ST − page-only critic	HTML repair	238	0.40	[0.09, 0.71]	0.021
Value ordering − error ordering, same checkpoint	HTML repair	238	0.62	[0.27, 0.96]	0.004
SGCA-S − multi-turn GRPO	Chart repair	582	0.75	[0.21, 1.30]	0.021
SGCA-T − multi-turn GRPO	Chart repair	582	1.07	[0.40, 1.73]	0.012
SGCA-ST − SGCA-S	Chart repair	582	0.94	[0.33, 1.55]	0.013
SGCA-ST − multi-turn GRPO	HTML repair	238	1.37	[1.06, 1.68]	–
SGCA-ST − SGCA-T	HTML repair	238	0.53	[0.22, 0.83]	–
Value ordering − SGCA-ST	HTML repair	238	0.47	[0.15, 0.78]	–
SGCA-ST − multi-turn GRPO	Chart repair	582	1.69	[1.05, 2.33]	–
SGCA-ST − SGCA-T	Chart repair	582	0.63	[0.06, 1.19]	–
SGCA-S − AT-RL	Design2Code	447	1.22	[0.41, 2.02]	–
SGCA-S − ReLook	Design2Code	447	0.91	[0.05, 1.77]	–
SGCA-ST − WebGen-Agent	HTML repair	238	1.12	[0.78, 1.46]	–
SGCA-ST − MURPHY	HTML repair	238	0.77	[0.46, 1.09]	–
SGCA-ST − multi-turn GRPO, seed 1	HTML repair	238	1.35	[0.93, 1.77]	–
SGCA-ST − compute-matched multi-turn GRPO, seed 1	HTML repair	238	0.71	[0.27, 1.15]	–

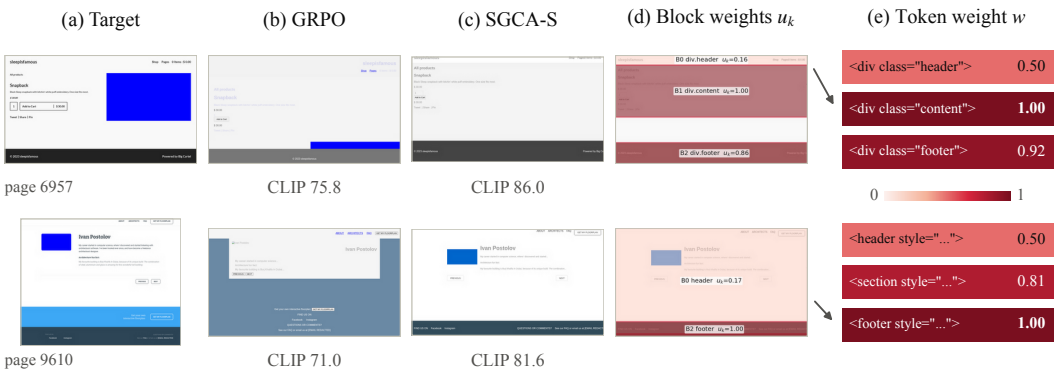

Figure A1: Spatial credit on two further Design2Code pages. Panels as in Figure 4.

Table A6: Final CLIP per seed and cost relative to GRPO of the same task. GPU time includes the amortized critic warm-up and chart SFT. Last HTML row: value-ordering checkpoints with error order.

		CLIP ↑			Cost relative to GRPO		
Task	Condition	Seed 1	Seed 2	Seed 3	GPU time	Peak memory	Latency
Design2Code	GRPO	71.07	71.30	71.14	1	1	1
	AT-RL	71.58	71.47	71.41	1.29	1.12	1.02
	ReLook	71.85	71.82	71.71	2.08	1.12	1.03
	SGCA-S	72.88	72.57	72.65	1.22	1.05	1.03
	Block-shuffled	71.04	71.15	71.14	1.22	1.05	1.02
HTML repair	Multi-turn GRPO	83.89	83.87	84.01	1	1	1
	SGCA-S	84.57	84.33	84.35	1.20	1.05	1.04
	SGCA-T	84.74	84.77	84.80	1.58	1.08	1.06
	SGCA-ST	85.24	85.38	85.27	1.78	1.13	1.09
	WebGen-Agent	84.22	84.20	84.10	2.06	1.12	2.01
	MURPHY	84.53	84.57	84.47	2.08	1.13	2.00
	Return-to-go, no critic	84.90	84.94	84.66	1.23	1.05	1.03
	Page-only critic	84.93	84.79	84.97	1.71	1.09	1.07
	Value ordering	85.87	85.69	85.73	1.93	1.16	1.41
	with error order	85.04	85.25	85.14	–	–	1.12
Chart repair	Multi-turn GRPO	76.07	75.96	76.04	1	1	1
	SGCA-S	76.90	76.74	76.68	1.21	1.05	1.03
	SGCA-T	77.04	77.09	77.12	1.63	1.08	1.06
	SGCA-ST	77.67	77.83	77.64	1.84	1.12	1.09

Table A7: Paired differences by page complexity (95% intervals). Nodes: range of visible DOM nodes.

Comparison	Bucket	Nodes	Pages	ΔCLIP	95% CI	ΔMSE	ΔLPIPS
SGCA-S − GRPO (Design2Code)	Low	7–43	152	0.54	[0.09, 0.99]	−0.67	−2.25
	Medium	44–106	147	1.37	[0.15, 2.58]	−1.13	−3.95
	High	107–347	148	2.71	[1.36, 4.07]	−1.82	−7.19
SGCA-ST − multi-turn GRPO (HTML repair)	Low	9–38	79	0.61	[0.13, 1.10]	−0.04	−1.04
	Medium	39–92	80	1.40	[0.87, 1.92]	−0.12	−3.20
	High	93–286	79	2.10	[1.54, 2.66]	−0.51	−5.24

Table A8: Multi-turn GRPO at the training cost of SGCA-ST (HTML repair, seed 1). GPU time: full critic warm-up of the seed included.

Run	Groups	GPU time (ratio)	CLIP ↑	MSE ↓	LPIPS ↓
Multi-turn GRPO	400	1	83.89	4.02	17.71
Multi-turn GRPO, continued	835	2.08	84.53	3.92	17.23
SGCA-ST	400	2.07	**85.24**	**3.86**	**14.74**

Fault localization. One fault is injected into one block of a development page: replaced text, inverted color, larger font, extra padding, or a wider box, 20 pages each. The original render is the target, and the frozen attribution pipeline scores the blocks of the faulty page. A page offers 12 to 38 candidate blocks, 25 on average (Table A9).

Table A9: Localization of injected single-block faults (100 pages, 95% intervals). Top-1: in %. Uniform: ties broken at random.

			Top-1 by fault type				
Token weights	Top-1 ↑	MRR ↑	Text	Color	Font size	Padding	Width
Correct node-to-token map	**67** [58, 76]	**0.70** [0.61, 0.78]	**70**	**60**	**80**	**65**	**60**
Block-shuffled	1 [0, 3]	0.11 [0.09, 0.14]	0	0	0	0	5
Uniform	4.2 [3.8, 4.5]	0.15 [0.14, 0.16]	4.4	3.8	4.5	4.0	4.1



Figure A2: Close-ups of the edits in Figure 6. Target, before, and after share screen coordinates. CLIP on the 0–1 scale.

Chart components. Table A10 gives the four F1 scores behind the low-level score. Both credits raise text, layout, and color over multi-turn GRPO, and SGCA-ST is the best on all three. Spatial credit helps layout most, and temporal credit also raises the chart-type score.

Table A10: ChartMimic component scores (F1). Their mean is the low-level score of Table 4.

Method	Text ↑	Layout ↑	Type ↑	Color ↑
No-edit (shared draft)	72.8	62.4	88.7	66.2
Multi-turn GRPO	73.9±0.8	66.4±0.5	90.2±1.0	67.8±0.5
SGCA-S	75.2±0.6	70.6±0.4	89.7±0.8	69.9±1.1
SGCA-T	76.5±0.3	69.9±0.5	**91.7**±0.6	70.4±0.5
SGCA-ST	**77.3**±1.0	**73.7**±0.3	91.4±0.8	**73.2**±0.2