URBANE:

URBAN REASONING AND (CITY) BLOCK ADJUSTMENT VIA NATURAL-LANGUAGE EDITING

Tanner Waltz, Julio Vera, and Daniel Aliaga

ICPR 2026

Presented by Daniel Aliaga

PURDUE UNIVERSITY
Department of Computer Science

Problem: High-level urban design intent has no direct geometric operator

Prior Approaches:

- Manual/CAD approaches
  - Time consuming and expertise-intensive
- (AI-based) layout generation
  - Generates plausible layouts, but offers limited control over editing existing layouts

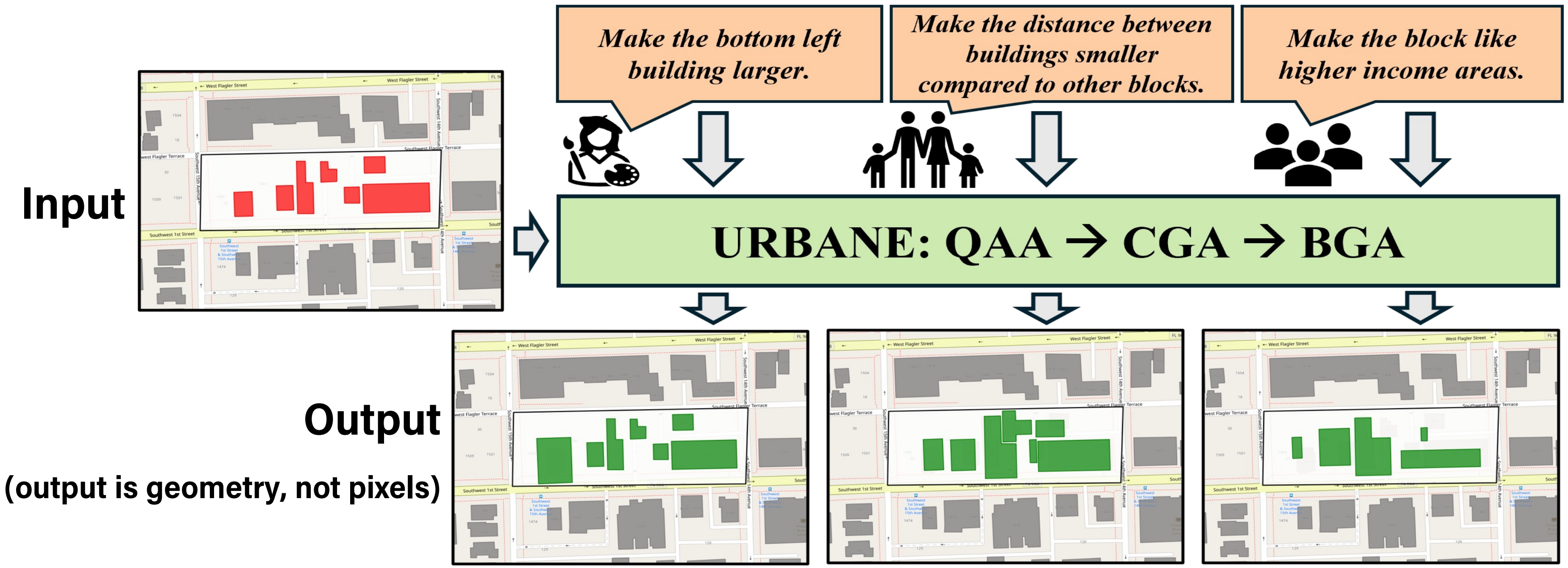
Observation:

- A natural-language interface can request complex edits without requiring users to specify low-level geometric operations

Our Solution:

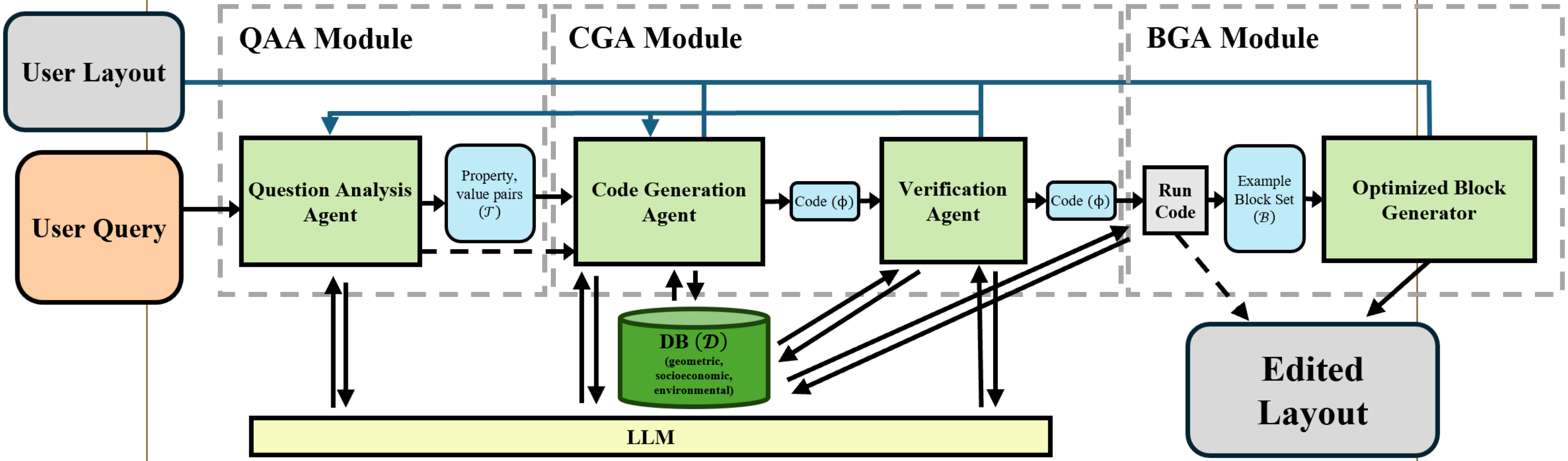
- *1. interpret intent from natural language*
- *2. structure into measurable targets*
- *3. produce valid polygonal geometry of edited urban layouts*

URBANE: Urban Reasoning and Block Adjustment via Natural-Language Editing



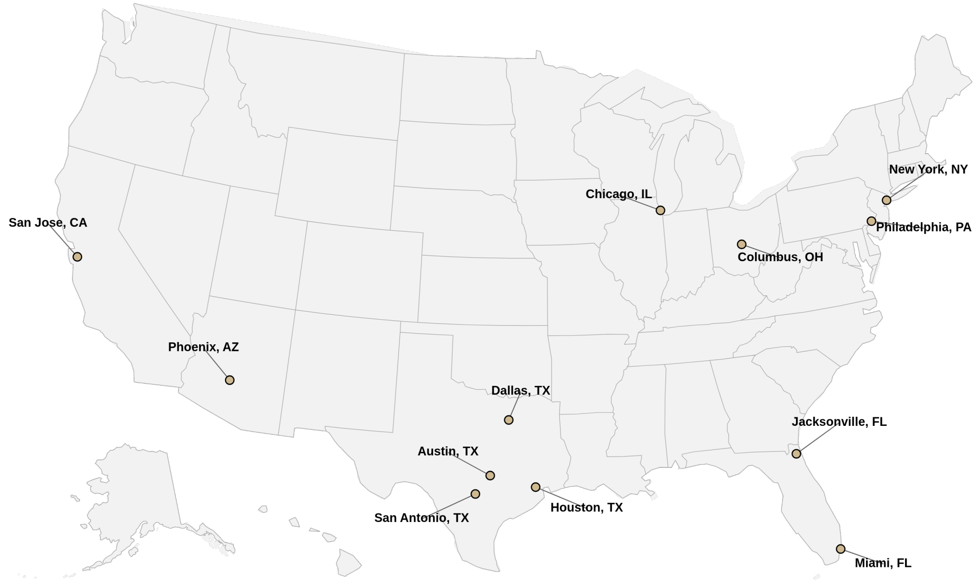
Example (input)	Request Level	Result (output)
"Make the bottom-left building larger."	Building-level	Scale one polygon
"Make the distance between buildings smaller compared to other blocks."	Geometric Block-level	The entire layout moves toward a geometric target
"Make the block like higher-income areas."	Semantic Block-level	Geometry moves toward real blocks associated with the requested attribute

URBANE has three main modules



112,182 real-world city blocks ground contextual editing

12 U.S. cities spanning diverse urban forms



Data layers

- **OpenStreetMap**: Block boundaries and spatial context
- **Microsoft Building Footprints**: Building polygons
- **Climate and Economic Justice Screening Tool (CEJST)**: Socioeconomic and environmental indicators

Feature groups

- **Geometry**: area, spacing, orientation, medial-axis distance
- **Context**: city, county, parcel count, layout category
- **Socio-environmental**: income, education, employment, flood/fire risk, pollution

QAA: Question Analysis Agent

1. Setup problem

- Receive user request in natural language r
- Receive urban block layout $\mathcal{L} = (B, \{b_i\}_{i=1}^{n}, M)$
  - B: block-boundary polygon
  - b_i: building polygons
  - M: medial axis used for spatial reasoning

✓ Buildings remain inside the boundary

✓ Buildings do not overlap

✓ Topological consistency is preserved

2. Classify request

- Building-level
  - "Rotate the top-right building by 45 degrees"
  - $(r, \mathcal{L}) \rightarrow \phi_{\text{bldg}} \rightarrow \text{execution} + \text{visual verification} \rightarrow \mathcal{L}'$
- Block-level
  - "Make the buildings farther apart compared to other blocks"
  - $r \rightarrow T \rightarrow \phi_{\text{filter}}(D) \rightarrow \mathcal{B} \rightarrow \text{optimization} \rightarrow \mathcal{L}'$

1. Request Classifier (LLM)

$f_{\text{type}}(r) \in \{building\!-\!level,\ block\!-\!level\}$

"Make the bottom left building larger." → **building-level**
"Make the distance between buildings larger compared to other blocks." → **block-level**
"Make the block like higher income areas." → **block-level**

3. Feature Tuple Extractor (for block-level)

- Extract relevant property fields
- Determine the requested value or direction for each property

2. Feature Tuple Extractor (LLM)

(if $f_{\text{type}}(r) = block-level$)

Property p_j	Value v_j	Property p_j	Value v_j
Building area	larger	Inter-building spacing	higher
Income	higher	…	…
…	…	…	…

CGA: Code Generation Agent

1. Building-level request

- **Output**:
  - Executable geometry-editing function ϕ_{bldg}
- **Example**:
  - ϕ_{bldg}: directly edits the specified building or buildings

2. Block-level request

- **Outputs**:
  - Natural-language subquery q
  - Relevant feature set $\{h_k\}$
  - Python predicate ϕ_{filter}
- **Examples**:
  - q: “blocks with large average inter-building distance”
  - $\{h_k\}$: mean pairwise building distance
  - ϕ_{filter}: select blocks above a chosen percentile

3. Retrieve examples for block-level requests

- $\mathcal{B} = \{L_j \in D \mid \phi_{\text{filter}}(L_j) = \text{True}\}$

CGA, Part 2: Verification Agent (VEA)

Building-level verification

1. Execute ϕ_{bldg}
2. Check that code returns a valid layout
3. Render original and edited layouts side by side
4. A multimodal LLM (MLLM) judges whether the visual change satisfies the request
5. Failed result returns diagnostic feedback to CGA

Block-level verification

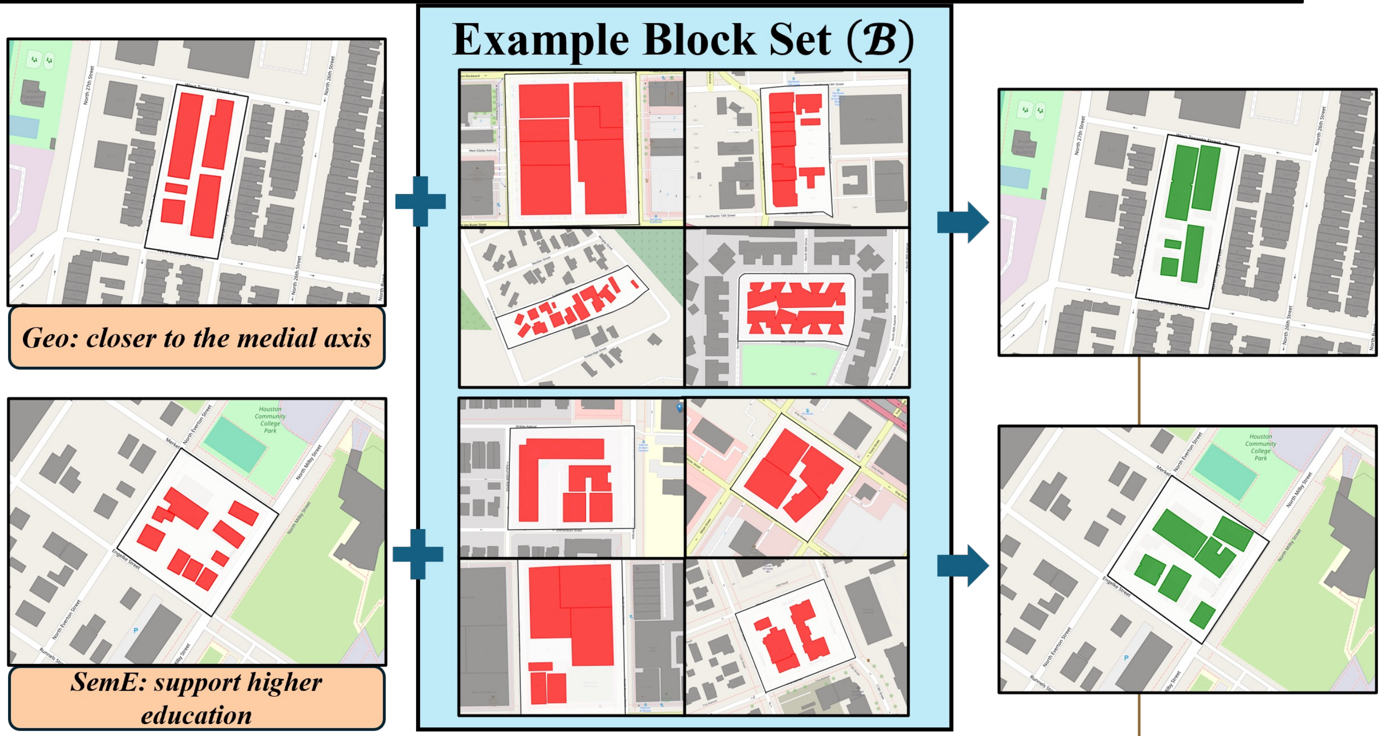
1. **Test on 10% random sample** $(\mathbf{0.1}N)$
   1-minute timeout
   ↓
2. **Error checking**
   execution validity
   result-set bounds
   valid block identifiers
   ↓
3. **Full database execution**

4. Maximum 5 iterations; fresh conversation after repeated failure
5. Full-database fallback when sampling repeatedly returns no examples

BGA: Block Generation Agent

Block-level four-stage process

1. **Shape parameterization**: Fit each building to its best {I, L, T, U, H} representation using Intersection-over-Union (IoU)
2. **Preprocessing**: Repair or remove overlaps and buildings extending beyond the block boundary
3. **Layout typing**: Classify as one of four organized types or "other"
4. **Optimization:**
   a. **Building-count adjustment**: Add or remove buildings to match target density
   b. **Geometric optimization**: Use CMA-ES (Covariance Matrix Adaptation Evolution Strategy) to optimize width, height, position, and rotation



- F: layout features
- μ, Σ: example-set mean and covariance
- Δ: edit parameters

- **Mahalanobis distance**: accounts for covariance among layout features
  - $D_M(x, \mu) = (x - \mu)^\top \Sigma^{-1} (x - \mu)$
- **Objective function**
  - $\mathcal{J}(\Delta) = D_M(\mathcal{F}(\mathcal{L}'(\Delta)), \mu)$

Evaluation

Category	*Count*	*Example*
Bldg — building-level	5	Rotate the top-right building 45°
Geo — geometric block-level	10	Make buildings farther apart
SemE — directly relevant features available	7	Support higher employment
SemH — directly relevant features removed	7	Same semantic requests, but directly relevant columns removed

29 total questions

CGA metric

- **Bldg**: The edit is correctly applied after validation by the VEA
- **Geo, SemE, SemH**: the retrieved set must shift the target property in the requested direction
- $\bar{p}(\mathcal{B}) > \bar{p}(D) \quad \text{or} \quad \bar{p}(\mathcal{B}) < \bar{p}(D)$

BGA metric (Block-level questions only)

- Higher is better; percentage normalization accounts for differing initial distances
- $\mathrm{D}_{\mathrm{orig}}$: distance from the input to the example-set distribution
- $\mathrm{D}_{\mathrm{edit}}$: distance from the edited layout to the example-set distribution
- $100 \times \frac{\mathrm{D}_{\mathrm{orig}} - \mathrm{D}_{\mathrm{edit}}}{\mathrm{D}_{\mathrm{orig}}}$

CGA Results: Success Rate by Request Category

Main CGA Results

SUCCESS RATE (%)	Bldg	Geo	SemE	SemH
Claude Haiku 4.5	90.38%	94.63%	75.61%	56.76%
GPT-5 Mini	94.85%	97.37%	77.94%	58.81%
Grok Code Fast 1	—	99.16%	82.80%	65.43%
Gemini 2.5 Flash	85.04%	99.33%	85.71%	68.97%

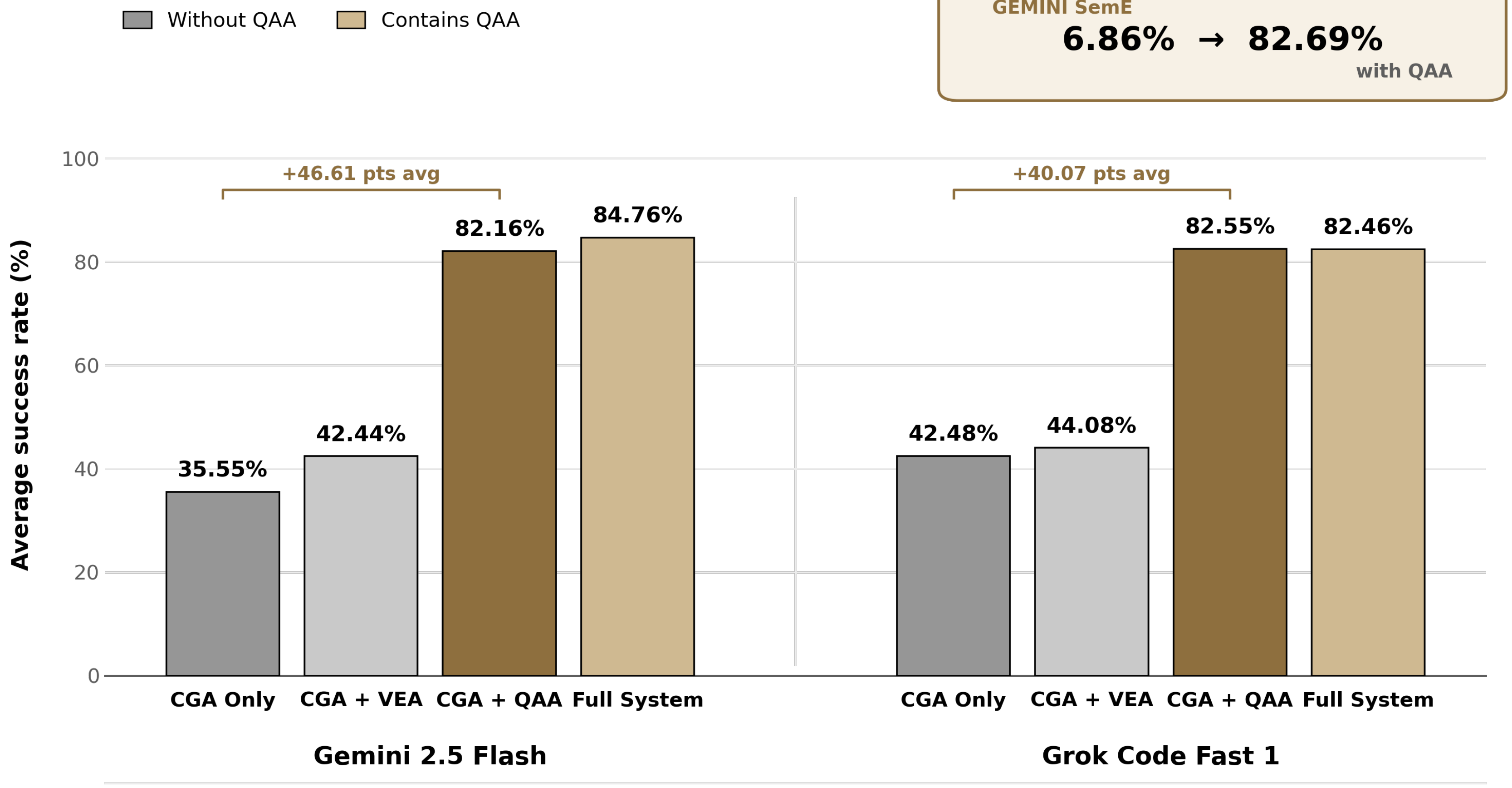
— Grok Code Fast 1 was not evaluated on Bldg.

Darker cells indicate higher success.

QAA and VEA Results: Measured by Ablation

Ablation Studies

GEMINI SemE
6.86% → 82.69%
with QAA



Bars show the Table 2 average across Geo, SemE, and SemH.

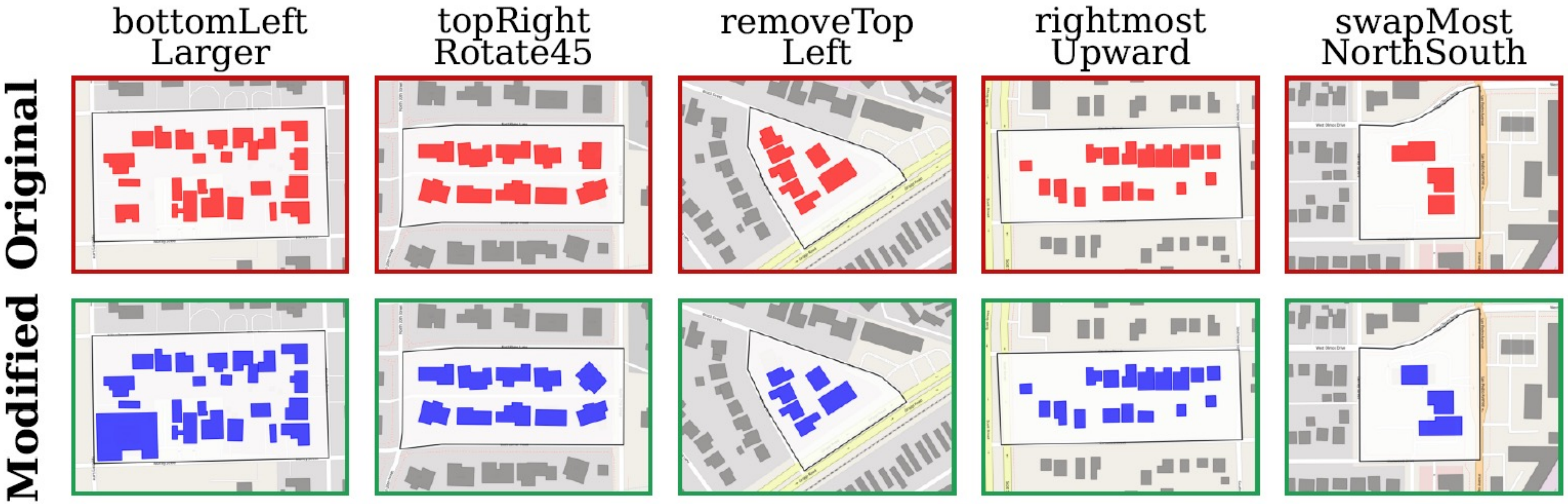
BGA Results: quantified over Geo, SemE, and SemH

- Metric used: percent reduction in Mahalanobis distance to the example-set distribution

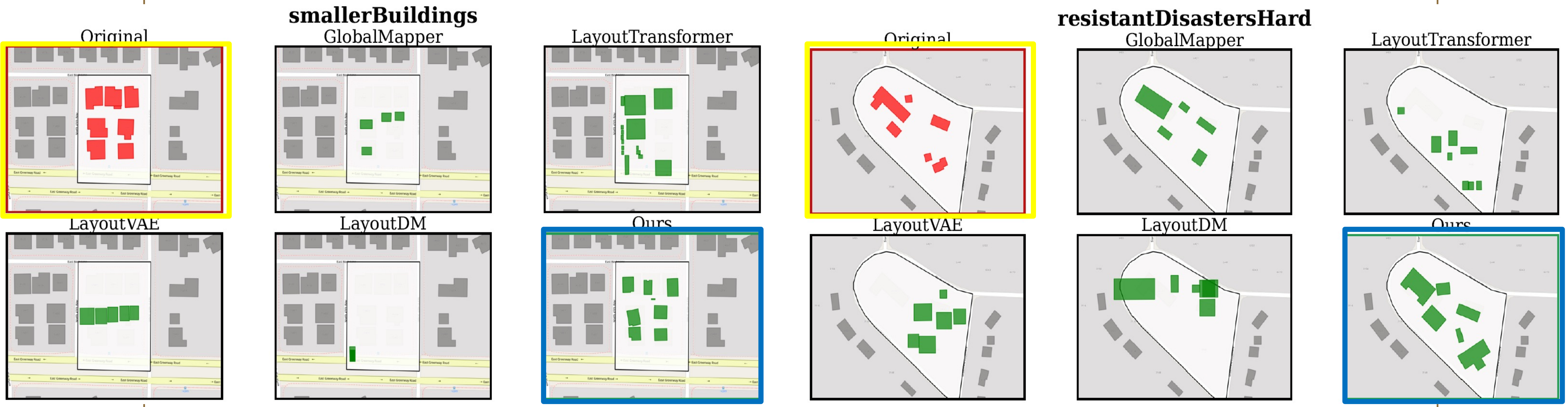
Method	Geo	SemE	SemH
BGA	49.95	53.74	54.47
LayoutDM	33.33	27.48	28.50
Pure LLM editing	1.79	1.14	1.43
LLM given target statistics, without BGA	4.62	2.46	2.83

- BGA exceeds the strongest comparison by 16.62-26.26 percentage points
- **Overall, BGA helps by 49.95 to 54.47%**

Qualitative Examples: Building-Level



Qualitative Comparison: Block-Level Edits



Dataset Composition and Schema

Panel A: Dataset composition

City	Blocks	City	Blocks
Houston, TX	17,253	Chicago, IL	14,291
New York, NY	13,017	Phoenix, AZ	12,651
San Antonio, TX	11,912	Dallas, TX	9,223
Philadelphia, PA	8,798	Jacksonville, FL	8,771
Columbus, OH	5,938	Austin, TX	5,528
San Jose, CA	3,298	Miami, FL	1,502
		Total	**112,182**

Panel B: Principal database feature groups

Feature group	Representative attributes
Geographic and administrative	State, county, and city identifiers.
Geometric and layout	Building coverage; building-area statistics; distance to the block boundary; pairwise building distances; layout category; parcel count; and block, building, and medial-axis geometry.
Environmental and climate risk	Diesel particulate-matter exposure; expected agricultural and building loss rates; 30-year flood and fire risk; and impervious-surface or cropland coverage.
Socioeconomic	Household financial burden; income and poverty measures; education indicators; and unemployment.
Housing and infrastructure	Pre-1960 housing; homes lacking a kitchen or indoor plumbing; traffic proximity and volume; and proximity to hazardous-waste sites.
Health and demographics	Low life expectancy; expected population loss; total population; and population by age group.

Public dataset

doi:10.5281/zenodo.19931111

Geometric evaluation prompts

Question	Shorthand
Geometric Questions (Geo)	
Make the block have larger buildings compared to other blocks.	largerBuildings
Make the block have buildings with smaller area compared to other blocks.	smallerBuildings
Make the distance between buildings larger compared to other blocks.	largerDistances
Make the distance between buildings smaller compared to other blocks.	smallerDistances
Make the block more covered by buildings compared to other blocks.	moreCovered
Make the block less covered by buildings compared to other blocks.	lessCovered
Make the buildings closer to the street compared to other blocks.	closerToStreet
Make the buildings farther from the street compared to other blocks.	fartherFromStreet
Make the buildings closer to the medial axis compared to other blocks.	closerToMedialAxis
Make the buildings farther from the medial axis compared to other blocks.	fartherFromMedialAxis

Building-level evaluation prompts

Structural Questions (Struct)	
Make the bottom left building much larger.	bottomLeftLarger
Rotate the top right building by 45 degrees.	topRightRotate45
Remove the top left building.	removeTopLeft
Move the rightmost building slightly upward.	rightmostUpward
Swap the northernmost building with the southernmost building.	swapMostNorthSouth

Semantic evaluation prompts

- **SemE:** directly relevant database features available
  SemH: the same prompts with those features removed

Question	Easy Shorthand	Hard Shorthand
Semantic Questions (Sem{E, H})		
Make this block have an older population.	olderPopulationEasy	olderPopulationHard
Make this block more resistant against natural disasters.	resistantDisastersEasy	resistantDisastersHard
Make the block support higher employment.	higherEmploymentEasy	higherEmploymentHard
Make the block support higher education levels.	higherEducationEasy	higherEducationHard
Make the block healthier.	healthierEasy	healthierHard
Make the block wealthier.	wealthierEasy	wealthierHard
Make the block more like Miami.	lookMiamiEasy	lookMiamiHard

URBANE: THREE TAKEAWAYS

1. **Decompose intent**
   QAA and CGA transform natural language into executable edits or data-grounded target sets.
2. **Verify and optimize**
   VEA catches failures; BGA returns valid polygonal layouts that move toward the retrieved examples.
3. **Strong empirical results**
   CGA success: 68.97–99.33%, depending on request difficulty
   BGA improvement: 49.95–54.47% reduction in distance to the target example set

Current scope: 2D building footprints and twelve U.S. cities
Future work: vegetation, transportation infrastructure, 3D buildings, and broader geographic coverage

Questions?