Semantic image editing for reshaping architecture of power: Lesson learned from selected cases

Dominik FILIPIAK,
Adam Mickiewicz University, Poznań, Poland
Perelyn, Warsaw, Poland
University of Innsbruck, Innsbruck, Austria
df@amu.edu.pl

Julita SOBICZEWSKA,
julita.sobiczewska@gmail.com

Abstract

Objectives We explore an evaluation scheme for assessment of generative computer vision models in architecture-related tasks with a focus on text-conditioned image editing for use cases relating to *architecture of power*. It is an umbrella term for building ranging from Socialist Realism to Post-War Modernism. While some of them can be considered landmarks on former Eastern Bloc countries, they often lack modern features, such as accessibility. With a recent progress in generative vision, the diffusion pipelines can be used to reimagine such buildings with pictures, which may later provide a blueprint for transforming such sites. **Prior work** While an intense effort can be observed in image generation models (including semantic image editing) and their applications (such as architecture), evaluating domain-specific benchmarks is still cumbersome. The case of architecture of power carries unique challenges, as it is a domain rather underrepresented in the publicly available datasets on which many models are pretrained. **Results** We present selected results of our evaluation schema for assessing generative vision models for various tasks related to improving mid-20th century architecture, which consist of taxonomy of tasks. We also demonstrate the proposed approach on a several state-of-the-art text- and image-conditioned diffusion models and pipelines (such as DiffEdit, Kandinsky, or ControlNet) for selected buildings in Warsaw, Cracow, Riga, and Bucharest. **Implications** While the presented evaluation scheme is rather intended to be used by researchers, the results of such an assessment can be used to select models most suitable for the architecture and urban planning communities. Since we focus on text-conditioned models, they can be used by general audience to help reimagining the buildings according to their need.

Keywords: semantic image editing, architecture of power, sustainability, evaluation, benchmark.

1. Introduction

Generative computer vision models are evolving rapidly, providing new capabilities in image generation or image editing. In architecture, image generation models can be used to transform and adapt existing structures to the current needs of residents and aesthetic changes considering sustainable and inclusive architecture. This work focuses on text-conditioned image editing for use cases relating to architecture of power, a term referring to the buildings ranging from Socialist Realism to Post-War Modernism. Such buildings are ubiquitous in regions such as former Eastern Bloc countries. While some of them can be considered landmarks on former Eastern Bloc countries, they often lack modern features, such as accessibility. Advances in generative vision, particularly in image editing pipelines, are opening new possibilities for reimagining and transforming existing buildings. Relying on social feedback, we aim to evaluate and compare the

effectiveness of different models and pipelines (implemented processes that use models to complete a specific task) in transforming architecture of power, with a focus on modernization and improving accessibility.

Most computer vision models have been trained on datasets that were created to capture a wide variety of images including: faces, cartoons, landscape, or art, rather than architectural photos. This creates a gap in their ability to understand and generate high-quality architectural images, which require the knowledge of spatial design, structural building elements and understanding of relevant architectural vocabulary. To address this limitation, our study compares models that use additional mask image as input, where the parts of the image to be modified are indicated, with models that perform modifications without mask image. By applying these two approaches, our goal is to understand how the selected editing methods differ when given either a text prompt and source image or a combination of a text prompt, source image, and mask image.

This article presents an initial case study, applying state-of-the-art generative computer vision models to explore their potential for reshaping the architecture of power. The findings from the research aim to provide insights and guide the design of further experiments in future research. The study used images from four cities: Bucharest, Cracow, Riga, and Warsaw, where numerous examples of architecture of power buildings can be found. To conduct our experiments, we selected iconic examples of this style, including House of Free Press, People's Palace, Congress building (Sala Palatului), National Theatre, Nowa Huta Administration Centre, Nowa Huta Museum, Hotel Cracovia, Latvian Academy of Sciences, Communist Party Headquarters, Riga Technical University and Red Riflemen Square, The Palace of Culture and Science, The Polish United Workers' Party Central Committee building, and Hotel Victoria. All photos were taken by members of our project team.

2. Related work

The field of generative computer vision in image editing is characterized by rapid progress and continuous development of new models and pipelines. As a result, newer model architectures and training methodologies are frequently introduced, offering better performance, improved semantic control, more precise image modifications and expanded applicability in various fields, including architecture. Prior research has explored the use of generative models for architecture, including the application of Stable Diffusion to generate high-quality images of historical arcade facades [1], or using generative computer vision for architectural design ideation [2]. Among the proven pipelines for image editing are ControlNet, DiffEdit, and Kandinsky.

2.1. Models

ControlNet [3] enhances image editing by providing additional control over various aspects of the output, such as human pose, or user sketching. It works by conditioning the model on additional information such as edge maps, segmentation masks, or depth maps, allowing for precise control over architectural edits. The ControlNet structure is designed for a wider range of conditions, allowing applications to be adjusted to specific requirements.

Proposed by [4], DiffEdit (diffusion-based semantic image editing with mask guidance) uses diffusion models for image editing, while maintaining similarity to the original input photo. Using an iterative noise-based process, DiffEdit allows new image elements to be integrated into the input photo, whether adding new structures or modifying existing ones.

DiffEdit can generate a mask to highlight regions that require modification if one is not provided by the user.

Kandinsky [5] includes the CLIP (Contrastive Language-Image Pretraining) model [6] into its architecture, enabling a more efficient mapping between text and image embedding.

This enhanced mapping strengthens the alignment of text and image, and when utilized during training with text embeddings, it produces higher-quality generated images.

Kandinsky 2.2 is an improvement of the previous model, replacing the image encoder with a larger *CLIP-ViT-G* model, resulting in better output quality.

2.2. Pipelines

DiffEdit and ControlNet pipelines use diffusion models for the image editing task. For this reason, we chose two stable diffusion models: *Stable Diffusion 2.1* [7] and newer *Stable Diffusion XL* [8], which differ in the quality level of their outputs. Meanwhile, the Kandinsky pipeline uses its own model: *kandinsky-2-2-decoder*. While all these pipelines and models have demonstrated strong performance across various domains, we aimed to evaluate their applicability and effectiveness specifically within the context of architecture.

Pipelines allow for both inpainting and image-to-image tasks. Inpainting is distinguished by editing specific areas of the image (mask image) while leaving the rest of the image untouched, while image-to-image modifies the entire image based on a given text description. To enable inpainting without the need for pre-generated mask images, we integrated the Segment Anything Model (SAM) [9], developed by Meta AI. SAM is a universal model for unsupervised semantic segmentation. By

providing positive and negative point coordinates of the input image, SAM generates masks that can be used with the image editing model, allowing for more precise control over the areas to be modified.

When applying image editing tasks using generative computer vision models to the problem of architecture transformation, we often need to segment specific building elements, e.g. windows, doors, facades. Thus, SAM aims to improve and partially automate the process of segmenting architectural elements.

3. Experiment setting

To effectively evaluate pipelines and models, we divided the benchmarks into four distinct tasks: *add*, *change*, *replace*, and *remove*. This categorization was intended to validate and understand the specific image modifications. By dividing the benchmarks into these four tasks, as summarized in Table 1, we could focus on how each model addresses different types of alterations, thus facilitating a comprehensive evaluation of their performance.

Table 1. List of evaluated benchmarks.

ID	Task	Full name	Used label
B1	*Add*	*Add park*	-
B2	*Add*	*Add ramp for people with disabilities*	Stairs
B3	*Add*	*Add flowers to the balcony*	Balcony
B4	*Add*	*Add flowers to the windows*	Window
B5	*Add*	*Add green roof*	Roof
B6	*Replace*	*Replace doors with automatic doors*	Doors
B7	*Replace*	*Replace square with playground*	Square
B8	*Change*	*Change elevation to clean*	Elevation
B9	*Change*	*Change façade panels*	Façade
B10	*Change*	*Change building materials to glass*	Building
B11	*Change*	*Change columns colour*	Column
B12	*Remove*	*Remove cars*	-

Source: own research

For the evaluation, we manually labelled a dataset of 110 images of the project buildings.

Each image was individually assessed and labelled according to the benchmarks that were applicable to the specific photo. This involved applying each of the 12 benchmarks (Table 1) to determine which modifications were relevant to the features present in each image. For example, if the image contained an unused area, a parking, or a street around a building, benchmark B1 was used. Similarly, benchmarks related to changes in specific building elements, e.g. windows, doors, and columns, were applied to images where these features were identifiable. This labelling process ensured that each image was accurately labelled, allowing precise evaluation of pipelines and models based on their ability to edit specific elements.

In the evaluation, we used several state-of-the-art pipelines and models, described in Section 2, including ControlNet, DiffEdit, and Kandinsky. These pipelines were selected based on our research into pipelines specifically designed for inpainting and image-to-image tasks, ensuring that they could effectively handle a range of various image editing scenarios. Most parameters across all pipelines were set to their default values. However, during implementation, we identified improved values for certain parameters and adjusted them accordingly, including: *negative_prompt*, and *strength*. The parameter *negative_prompt* has been set for ControlNet and Kandinsky pipelines with value low *quality, bad quality, unrealistic* as it significantly improved the performance of the pipelines. After testing, the *strength* parameter for the Kandinsky model was adjusted to 0.15 to better preserve the architectural features of the building in the original photo. The target prompts for each benchmark are shown in Table 2.

Table 2. Benchmark target prompts and positive/negative words

Benchmark ID	Target prompt
B1	A building surrounded by a lush park with trees, grass, walking paths, and benches, creating a natural and serene environment.
B2	A building entrance with a wheelchair ramp added alongside the stairs, ensuring accessibility.
B3	A building with vibrant flowers on the balconies, with planters filled with colorful blooms.
B4	A building with vibrant flowers on the windows, featuring colorful blooms and greenery in window boxes.
B5	A building with a green roof, covered in lush plants and grass, creating an eco-friendly space.
B6	A building with a sleek aluminum and glass automatic door, featuring a minimalist design and full-length glass panels.
B7	A public square with playground features, such as swings, slides, and climbing structures.
B8	A building with a sleek, clean elevation, featuring smooth, polished surfaces.
B9	A building with a facade featuring precast concrete panels with an exposed aggregate finish, adding texture and durability.
B10	A building with large glass walls, transparent and reflective panels, smooth surface, and modern design.
B11	A building with bright red columns, standing out vividly against the facade.
B12	A building with an adjacent area that is open and clear of vehicles, with no cars present.

Source: Own research

These pipelines were tested on two models: Stable Diffusion (SD2-1) and Stable Diffusion XL (SD-XL). SD2-1 was selected for its well-established performance and efficiency in generating high-quality images with relatively small computational demands. SD-XL, a more recent and significantly larger model, was chosen for its improved architecture, capable of producing images with greater detail, and resolution. By comparing these two models, we aimed to understand how newer, more complex architectures like SD-XL enhance performance over earlier iterations like SD2-1, particularly in tasks involving image-to-image and inpainting operations.

For benchmarks where masks were required to modify specific image elements, the Segment Anything Model (SAM) was applied. The model was chosen for its efficient segmentation capabilities, which allow it to generate accurate masks for a wide range of objects and regions without the need for task-specific training. Its ability to generalize across different scenarios made it well-suited for our benchmarks, where precise object-based modifications were necessary for tasks such as adding, replacing, and changing elements in the images.

SAM provided an effective method for masking specific parts of an image, allowing precise editing of target areas. Segment Anything model was not applicable to two benchmarks: B1 and B12, where there was no mask for changes, as can be seen in Table 1. Since these tasks involved modifications to areas surrounding the building, such as adding a park or removing cars, rather than to the building elements themselves, no masks were provided.

This combination of pipelines and models, along with the use of SAM when appropriate, provides a robust framework for evaluating image modifications across different tasks and benchmarks. Table 3 presents the combinations of all pipelines and models used in the experiment.

Table 3. Pipelines and models used in the evaluation.

Pipeline	Model	SAM
ControlNet	SD2-1	✓
ControlNet	SD2-1	-
ControlNet	SD-XL	✓
ControlNet	SD-XL	-
DiffEdit	SD2-1	✓
DiffEdit	SD2-1	-
DiffEdit	SD-XL	✓
DiffEdit	SD-XL	-
Kandinsky	Kandinsky	✓
Kandinsky	Kandinsky	-

Source: Own research

4. Results

In this section, we present sample results of our experiments, evaluating the performance of selected generative computer vision pipelines and models on defined benchmarks. For all 12 benchmarks, we discuss the performance of the models, analysing image modification accuracy, and the ability of the models to edit architectural elements, highlighting the strengths and limitations observed during the evaluation. Due to the large number of images, we have included photos of five benchmarks in the main text: B1 (add park), B3 (add flowers to the balcony), B6 (replace doors with automatic doors), B8 (change elevation to clean), and B12

(remove cars) from each category: *add*, *change*, *replace*, and *remove*, while the images for remaining examples are presented in the appendix A.

4.1. Benchmark B1 (add park)

Due to the nature of this benchmark, visual prompting solutions have been excluded.

Figure 1 depicts Riga Technical University and Red Riflemen Square, which turned out to be a surprisingly hard environment to add a park to for tested models. While it is hard to choose which one was the best, one can argue Kandinsky was the closest. DiffEdit with SD-XL reinterpreted the whole complex, but it was out of scope for this benchmark. Results are present in Table 5.

Fig. 1. Picture P61, an example input for benchmark B1 and B7.
Source: unLoc project pictures

Fig. 2. Picture P13, an example input for benchmark B2, B8, and B10.
Source: unLoc project pictures

Table 4. Example results for benchmark B1 (add park) with picture P61 (Figure 1).

Sample results	Sample results (with SAM)
 ControlNet (SD2-1)	 ControlNet (SD-XL)
 DiffEdit (SD2-1)	 DiffEdit (SD-XL)
 Kandinsky	

Source: Own research, unLoc project pictures

4.2. Benchmark B2 (add ramp for people with disabilities)

For all three models (ControlNet, DiffEdit, Kandinsky), there was at least one result with an added ramp for the benchmark B2. However, the overall results are various. In line with its tendency, DiffEdit SD-XL yields aesthetically pleasing results,

which are very invasive at the same time and alter the building. However, that ramp was arguably the most convincing one. Results are present in Table 6 (Appendix A).

4.3. Benchmark B3 (add flowers to the balcony)

Figure 2 pictures P88, which was among the ones used in this benchmark. Only one version of ControlNet handled this task for this picture, whereas it was not a problem for DiffEdit (except for SD-XL, which again changed the structure of the building). Typically, Kandinsky displayed problems with colours This time it has also problems with keeping the scale of added flowers right. Results are present in Table 5.

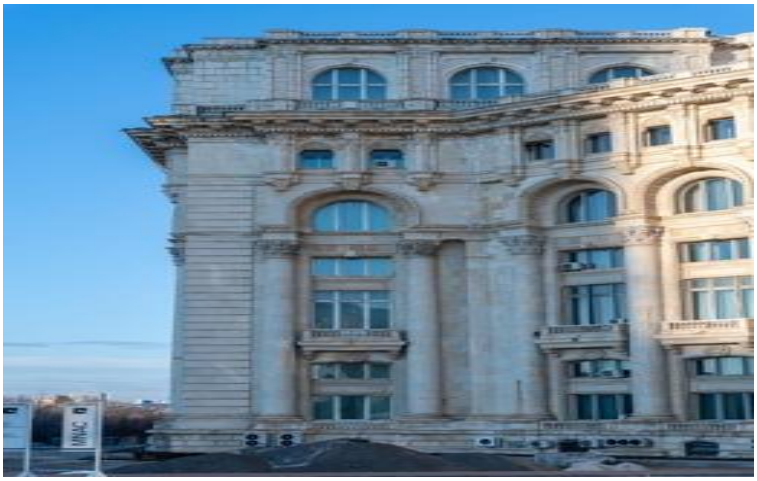
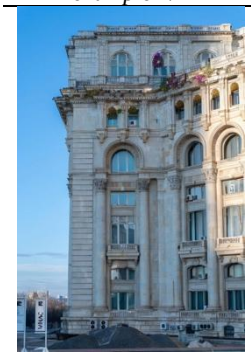
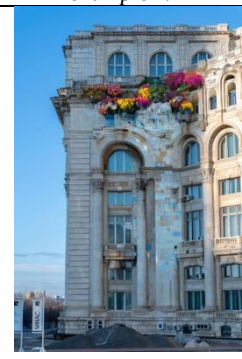
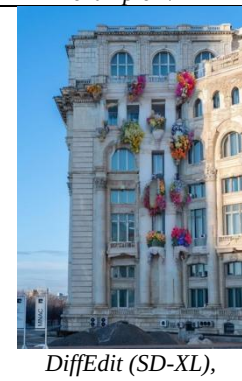


Fig. 3. Picture P88, an example input for benchmark B3.
Source: unLoc project pictures

Fig. 4. Picture P82, an example input for benchmark B6.
Source: unLoc project pictures

Table 5. Example results for benchmark B3 (add flowers to the balcony) with picture P88 (Figure 2).

Sample results		Sample results (with SAM)	
			
ControlNet (SD2-1), example 1/2	*ControlNet (SD-XL), example 1/2*	*ControlNet (SD2-1), example 2/2*	*ControlNet (SD-XL), example 2/2*
			
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD-XL), example 1/2*	*DiffEdit (SD2-1), example 2/2*	*DiffEdit (SD-XL), example 2/2*
			
Kandinsky, example 1/2		*Kandinsky, example 2/2*	

Source: Own research, unLoc project pictures

4.4. Benchmark B4 (add flowers to windows)

Benchmark B4 paired with picture P46 (*Figure 1*) poses an interesting task, as this building consists of numerous, (relatively) tiny windows. Indeed, it was challenging for the models to render proper flowers. Results are present in Table 7 (Appendix A).

4.5. Benchmark B5 (add green roof)

Placing some green on the roof at the very top of the Palace of Culture and Science posed a challenging problem for the models which were not guided with visual prompting.

DiffEdit with SD2-1 and Kandinsky did it somewhat correctly, albeit the former lost some details (notably on the clock), whereas the latter added some unexpected colours. Typically, DiffEdit with SD-XL generated a very aesthetic yet very different building of a similar shape. Paired with SAM, all the models did the task correctly. Results are present in Appendix A.

4.6. Benchmark B6 (replace doors with automatic doors)

One of the entrances to Warsaw's former Stock Exchange building leads to a beautiful, yet wheelchair-unfriendly door. SAM-guided models performed much better at this task, as the unguided ones clearly had problems with spotting or rearranging the doors (except for DiffEdit). The models which failed at it displayed their typical flaws in line with previous results. Results are present in Table 6.

Table 6. Example results for benchmark B6 (change elevation to clean) with picture P82 (Figure 3).

Sample results	Sample results (with SAM)
 *ControlNet (SD2-1), example 1/2*	 *ControlNet (SD2-1), example 2/2*
 *ControlNet (SD-XL), example 1/2*	 *ControlNet (SD-XL), example 2/2*
 *DiffEdit (SD2-1), example 1/2*	 *DiffEdit (SD2-1), example 2/2*
 *DiffEdit (SD-XL), example 1/2*	 *DiffEdit (SD-XL), example 2/2*
 *Kandinsky, example 1/2*	 *Kandinsky, example 2/2*

4.7. Benchmark B7 (replace square with playground)

For this example, we revisit P61 (Figure 1). Only one version of guided ControlNet created a proper playground, contrary to three versions of DiffEdit (often at the cost of not preserving the building structure). Kandinsky yielded semi-correct pictures, but they yet again contain strange artefacts. Results are present in Appendix A.

4.8. Benchmark B8 (change elevation to clean)

Table 7 shows example results for benchmark B8 for picture P13 (Figure 3). Arguably, ControlNet and Kandinsky (both without SAM) were best at handling elevation refreshing.

Interestingly, the versions with SAM also changed the elevation colour/material, or even the whole structure. DiffEdit, while usually aesthetically pleasing, introduced prominent facade changes in all versions. Kandinsky without SAM yielded somehow correct results (despite introducing unnecessary artefacts), but the SAM version performed much worse and altered the facade as well.

Sample results	Sample results (with SAM)
	
ControlNet (SD2-1), example 1/2	*ControlNet (SD2-1), example 2/2*
	
ControlNet (SD-XL), example 1/2	*ControlNet (SD-XL), example 2/2*

DiffEdit (SD2-1), example 1/2

DiffEdit (SD2-1), example 2/2

DiffEdit (SD-XL), example 1/2

DiffEdit (SD-XL), example 2/2

Kandinsky, example 1/2

Kandinsky, example 2/2

4.9. Benchmark B9 (change façade panels)

We present results for this benchmark for Figure 4. In this example, the localisation is somewhat easy, as the whole visible building is the area of concern, whereas other buildings are not present. However, the concept of panels turned out to be tricky for the models. Results are present in the appendix.

4.10. Benchmark B10 (change building material to glass)

Sample benchmark B10 results are presented in the appendix for picture P13 (Figure 2).

We find the results from DiffEdit (SD-XL) to be the most interesting. As it turns out, SAM yielded questionable results and marked only part of the building, which affected the results of all models using it.

4.11. Benchmark B11 (change column colour)

The results for this benchmark paired with picture P11 displayed some intriguing properties of the model. While it was expected that some models would have trouble with correctly spotting the narrow columns, the notion of red colour, or even the colour itself turned out to be not trivial. Whether with SAM or not, DiffEdit with SD-21 changed the columns, but it appears to not understand the notion of colour. It was not a problem for SD-XL (which changed too much without visual prompts). SAM-paired ControlNet yielded good results with SD-21 and partially good for SD-XL. Kandinsky either changed the building to red without columns or rendered red columns, but with a mismatched scale. Results are present in the appendix.

4.12. Benchmark B12 (remove cars)

Moving cars from perspective was also surprisingly difficult. Only DiffEdit did it right, albeit either with an out-of-place replacement (SD2-1) or altering the building (SD-XL). Results are present in Table 7.

Table 7. Example results for benchmark B12 (remove cars) for Figure P34.

Sample results	Sample results (with SAM)
	
ControlNet (SD2-1)	ControlNet (SD-XL)
	
DiffEdit (SD2-1)	DiffEdit (SD-XL)
	
Kandinsky	

4.13. Discussion

Previous subsections revealed some key trends describing the pipelines with DiffEdit emerging as a particularly promising one. Paired with SD-XL, it often generated stunning visualisation – however, they often contained way too many changes. Therefore, SAM is rather a must in practical settings.

ControlNet, on the other hand, seems to work best with SD2-1. While its images are often less appealing than the ones from DiffEdit, the introduced changes seek to be more conservative and therefore more structure-preserving. Kandinsky, while

having its forte such as elevation cleaning, in general often introduced unwanted artefacts. In particular, Kandinsky seems to struggle with preserving colours and scale.

While SAM is in general a very versatile model for unsupervised segmentation, it is not perfect. By design, it tries to find a single instance belonging to the same semantic class as the selected points. However, for disconnected objects, such as windows, this is a problem, as this means one must provide visual prompts for every single window. Leaving such edge cases aside, SAM occasionally yielded improper masks for seemingly easy classes, such as facade. This biased the results of the pipeline and often rendered them unusable. SAM2 [10], the recent new version of Segment Anything might be an interesting direction for the project's future.

5. Summary

The goal of this case study was to evaluate text- and image-conditioned generative computer vision models for improving the architecture of power in terms of sustainability, accessibility, and modernization. In our experiments we used three pipelines: DiffEdit, ControlNet, and Kandinsky combined with two stable diffusion models (SD2-1 and SD-XL) and optional visual prompting (SAM), which resulted in 10 different combinations.

Based on the conducted experiments, we can summarise the paper with project-specific recommendations. From the tested models, we recommend using DiffEdit with SD-XL for future applications in the project. Depending on the use case, one might consider a version without SAM (for cases in which one can afford to alter a lot) and with SAM (when it's crucial to retain some elements of the input picture). Alternatively, ControlNet with SD2-1 offered a decent performance as well, whereas Kandinsky did not convince us in terms of its reliability.

While we know which model is the best among the tested ones, it does not mean that we can't improve any of them. Therefore, after choosing the model, we recommend taking some additional steps which likely will improve their performance. Beyond experiments on other parameters, we recommend creating a dataset of selected buildings with text descriptions and fine-tuning stable diffusion models using this dataset. This might improve the results, especially for the pipelines without visual prompting. One might also consider switching SAM to SAM2 [10], which authors argue outperforms its predecessor.

Additionally, it is important to consider newer models, such as Stable Diffusion 3 [11], which is already available as open-source and appears to be a promising candidate for further exploration.

Acknowledgements
We would like to thank all people engaged in the generative vision part of the unLoc project: Damian Mizera, Zuzanna Czernicka, Julia Malik, Hubert Orlicki, Jacek Złoczowski, Małgorzata Łuczyna, and others. The extended WP project report [12], which is complimentary to this publication, is also available.

This article is based on the research within the project "unLoc – Exploring the Synergy of Human and Machine Creativity in Architecture. Redesigning Urban Space through Machine Learning, Artistic Expression, and Community Collaboration". The project is conducted by Dr. Jacek Złoczowski and Dr. Małgorzata Łuczyna at the National Education Commission University in Cracow, Poland with partners: National School of Political and Administrative Studies SNSPA, Romania, represented by Professor Loredana Ivan, BEIA Consult Int. represented by Dr. Eng. George Suciu, Academy of Arts Latvia, represented by Dr. Liene Jakobsone.

Project released within call: BUILDING TRANSFORMATION CAPACITY THROUGH ARTS AND DESIGN: UNLOCKING THE FULL POTENTIAL FOR URBAN TRANSITIONS. A Joint Call within the European Union's Horizon 2020 research and innovation programme ERA-NET Cofund Urban Transformation Capacities (ENUTC) under Grant Agreement No. 101003758. Financing: National Science Centre Poland, Executive Agency for Higher Education, Research, Development and Innovation Funding, Romania, State Education Development Agency Republic of Latvia.

Appendix A. Additional benchmark examples

Fig. 5. Picture P46, an example input for benchmark B4 and B9.

Fig. 6. Picture P75, an example input for benchmark B5.

Fig. 7. Picture P11, an example input for benchmark B11.

Table 8. Example results for benchmark B2 (add ramp for people with disabilities) with picture P13.

Sample results	Sample results (with SAM)
	
ControlNet (SD2-1), example 1/2	*ControlNet (SD2-1), example 2/2*
	
ControlNet (SD-XL), example 1/2	*ControlNet (SD-XL), example 2/2*
	
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD2-1), example 2/2*
	
DiffEdit (SD-XL), example 1/2	*DiffEdit (SD-XL), example 2/2*
	
Kandinsky, example 1/2	*Kandinsky, example 2/2*

Table 9. Example results for benchmark B4 (add flowers to the windows) with picture P46 (Figure 3).

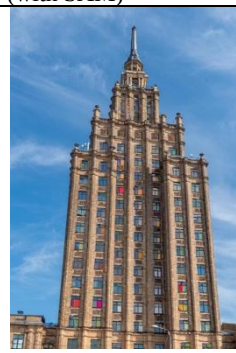
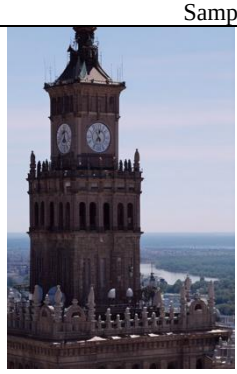
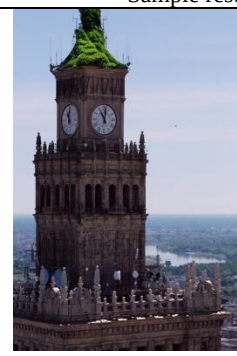
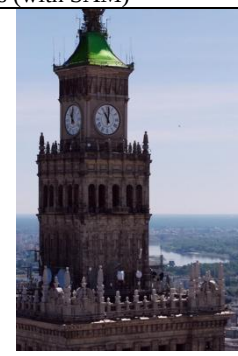
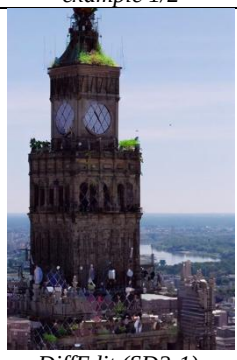
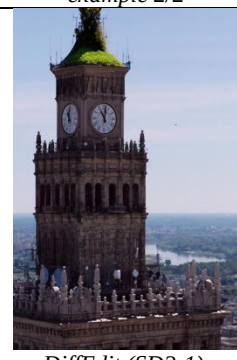
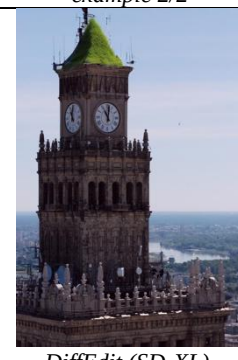
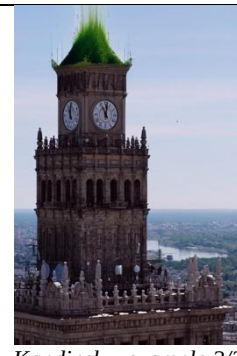
Sample results		Sample results (with SAM)	
			
ControlNet (SD2-1), example 1/2	*ControlNet (SD-XL), example 1/2*	*ControlNet (SD2-1), example 2/2*	*ControlNet (SD-XL), example 2/2*
			
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD-XL), example 1/2*	*DiffEdit (SD2-1), example 2/2*	*DiffEdit (SD-XL), example 2/2*
			
Kandinsky, example 1/2		*Kandinsky, example 2/2*	

Table 10. Example results for benchmark B3 (add flowers to the balcony) with picture P61 (Fig. 1).

Sample results		Sample results (with SAM)	
			
ControlNet (SD2-1), example 1/2	*ControlNet (SD-XL), example 1/2*	*ControlNet (SD2-1), example 2/2*	*ControlNet (SD-XL), example 2/2*
			
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD-XL), example 1/2*	*DiffEdit (SD2-1), example 2/2*	*DiffEdit (SD-XL), example 2/2*
			
Kandinsky, example 1/2		*Kandinsky, example 2/2*	

Source: Own research, unLoc project pictures

Table 11. Example results for benchmark B7 (replace square with playground) with picture P61 (Fig. 1).

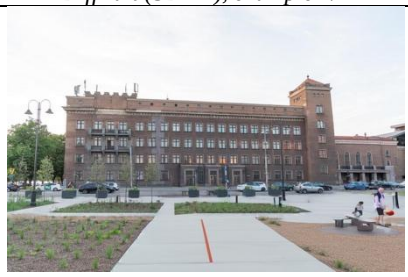
Sample results	Sample results (with SAM)
	
ControlNet (SD2-1), example 1/2	*ControlNet (SD2-1), example 2/2*
	
ControlNet (SD-XL), example 1/2	*ControlNet (SD-XL), example 2/2*
	
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD2-1), example 2/2*
	
DiffEdit (SD-XL), example 1/2	*DiffEdit (SD-XL), example 2/2*
	
Kandinsky, example 1/2	*Kandinsky, example 2/2*

Table 12. Example results for benchmark B7 (change elevation to clean) with picture P13.

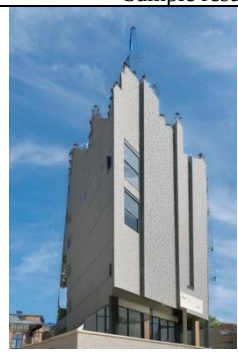
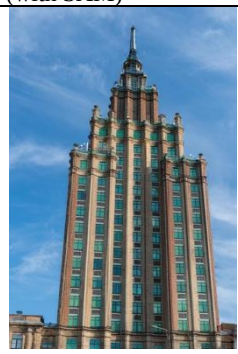
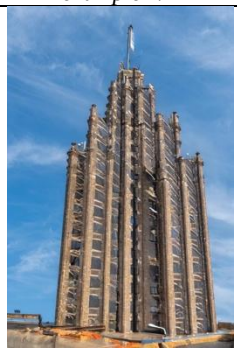
Sample results		Sample results (with SAM)	
			
ControlNet (SD2-1), example 1/2	*ControlNet (SD-XL), example 1/2*	*ControlNet (SD2-1), example 2/2*	*ControlNet (SD-XL), example 2/2*
			
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD-XL), example 1/2*	*DiffEdit (SD2-1), example 2/2*	*DiffEdit (SD-XL), example 2/2*
			
Kandinsky, example 1/2		*Kandinsky, example 2/2*	

Table 13. Example results for benchmark B10 (change elevation to clean) with picture P13.

Sample results	Sample results (with SAM)
	
ControlNet (SD2-1), example 1/2	*ControlNet (SD2-1), example 2/2*
	
ControlNet (SD-XL), example 1/2	*ControlNet (SD-XL), example 2/2*
	
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD2-1), example 2/2*
	
DiffEdit (SD-XL), example 1/2	*DiffEdit (SD-XL), example 2/2*
	
Kandinsky, example 1/2	*Kandinsky, example 2/2*

Table 14. Example results for benchmark B11 (change column colour) with picture P11.

Sample results	Sample results (with SAM)
	
ControlNet (SD2-1), example 1/2	*ControlNet (SD2-1), example 2/2*
	
ControlNet (SD-XL), example 1/2	*ControlNet (SD-XL), example 2/2*
	
DiffEdit (SD2-1), example 1/2	*DiffEdit (SD2-1), example 2/2*
	
DiffEdit (SD-XL), example 1/2	*DiffEdit (SD-XL), example 2/2*
	
Kandinsky, example 1/2	*Kandinsky, example 2/2*

References

[1] Z. Kuang, J. Zhang, Y. Huang and Y. Li, "Advancing Urban Renewal: An Automated Approach to Generating Historical Arcade Facades with Stable Diffusion Models," 2023. [Online]. Available: arXiv preprint arXiv:2311.11590.

[2] V. Paananen, J. Oppenlaender and A. Visuri, "Using text-to-image generation for architectural design ideation," *International Journal of Architectural Computing*, vol. 22, no. 3, p. 458–474, 2024.

[3] L. Zhang, A. Rao and M. Agrawala, "Adding Conditional Control to Text-to-Image Diffusion Models," 2023. [Online]. Available: https://arxiv.org/abs/2302.05543.

[4] G. Couairon, J. Verbeek, H. Schwenk and M. Cord, "DiffEdit: Diffusion-based Semantic Image Editing with Mask Guidance," *ICLR 2023 (Eleventh International Conference on Learning Representations)*, 2023.

[5] A. Razzhigaev and et al., "Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion," 2023. [Online]. Available: arXiv:2310.03502.

[6] A. Radford and et al., "Learning transferable visual models from natural language supervision," *International conference on machine learning*, p. 8748–8763, 2021.

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, "High-Resolution Image Synthesis With Latent Diffusion Models," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684 - 10695, 2022.

[8] D. Podell and et al., "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2307.01952.

[9] A. Kirillov and et al., "Segment Anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643.

[10] N. Ravi and et al., "SAM 2: Segment Anything in Images and Videos," 2024. [Online]. Available: https://arxiv.org/abs/2408.00714.

[11] P. Esser and et al., "Scaling rectified flow transformers for high-resolution image synthesis," *Forty-first International Conference on Machine Learning*, 2024.

[12] D. Filipiak and et al., "unLoc WP5 Report. Generative Computer Vision for Reimagining Architecture of Power," 2024.