Spatial Compositional Counterfactuals in Concept Bottleneck Models

Ran Eisenberg [1] **Ofir Lindenbaum** [1]

Abstract

Concept Bottleneck Models (CBMs) decompose images into semantic concepts to make predictions interpretable, but in doing so, they collapse the images' spatial structure. Counterfactual CBMs identify which concepts must change to alter the model's prediction, but they do not explain their spatial attribution. We introduce the Would-Have-Expected Concept Bottleneck Model (WHE-CBM), a spatial CBM that represents concepts and learns a counterfactual editor over those maps. Given a target object and desired class, the editor predicts a sparse continuous concept-logit delta whose sign, magnitude, and spatial support specify which semantic components must increase or decrease, and where, to flip the prediction. Across concept-controlled, label-free, and object-centric benchmarks, WHE-CBM improves counterfactual validity and sparsity over CF-CBM, localizes edit mass to the target object, and preserves non-target RoIs in multi-object scenes.

1. Introduction

Compositionality is the ability to build complex decisions from reusable components. Concept Bottleneck Models (CBMs) embody this principle: an image prediction is decomposed into semantic concept activations and a linear classifier over those activations (Koh et al., 2020). A change to these activations has a transparent downstream effect. However, visual concepts are rarely purely global. The same concept, such as stripes, wings, or wheels, can be reused across different objects and images. A compositional visual explanation should therefore identify not only which concept components matter, but also where they are instantiated and which object they affect.

This limitation is sharp in counterfactual explanation. Counterfactual CBMs identify concept changes that would alter a prediction (Dominici et al., 2025), but their edits are global concept-vector changes. Spatial CBMs localize concept evidence (Benou & Riklin-Raviv, 2025), and object-centric bottlenecks associate concepts with instances (Steinmann et al., 2025), but they do not train a model to generate editing suggestions that flip a specified object prediction while preserving the rest of the scene. Existing attribution methods atop CLIP-CBMs can explain which region supports a concept, but they do not learn spatial concept-space interventions that isolate a target instance.

Table 1. Comparison of CBM variants. WHE-CBM supports object-level reasoning, spatial concepts, and spatial counterfactual explanations. Symbols indicate feature support: ✓ supported, ✗ not supported, and ✗̸ partially supported.

Method	Concept expl.	Object level	CF expl.	Spatial concepts	Spatial CF
CBM (Koh et al., 2020)	✓	✗	✗	✗	✗
CF-CBM (Dominici et al., 2025)	✓	✗	✓	✗	✗
SALF-CBM (Benou & Riklin-Raviv, 2025)	✓	✗̸	✗	✓	✗̸
OCB (Steinmann et al., 2025)	✓	✓	✗	✗̸	✗
WHE-CBM	✓	✓	✓	✓	✓

We propose *Would-Have-Expected Concept Bottleneck Models* (WHE-CBM), a framework for spatial counterfactuals. WHE-CBM learns spatial concept maps, extracts RoI-aligned concepts for object-level decisions, and trains a class-conditional editor that outputs concept editing suggestions. The explanation is the edit itself: positive values indicate where a concept should increase, negative values indicate where it should be suppressed, and near-zero values identify concepts and locations that are not part of the intervention.

Unlike prior counterfactual CBMs, WHE-CBM does not merely indicate that a concept should increase or decrease globally; instead, it identifies where in the image the concept must change to affect a specific object-level prediction.

Our contributions are: (1) we formulate object-level counterfactual explanation as an intervention over spatial concept components; (2) we introduce WHE-CBM, which learns RoI-aligned spatial concept bottlenecks and a shared counterfactual editor; and (3) we evaluate counterfactual validity, sparsity, spatial locality, and object-level qualitative behavior across controlled and natural-image settings.

[1]Faculty of Engineering, Bar-Ilan University, Ramat Gan, Israel. Correspondence to: Ran Eisenberg <eisenbr2@biu.ac.il>, Ofir Lindenbaum <ofir.lindenbaum@biu.ac.il>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

2. Related Work

Concept bottlenecks. CBMs predict human-interpretable concepts before the task label, enabling transparent concept-level interventions (Koh et al., 2020). Post-hoc and label-free variants retrofit bottlenecks onto strong pretrained models or discover concepts with language supervision (Yuksekgonul et al., 2023; Oikarinen et al., 2023; Yang et al., 2023; Oikarinen & Weng, 2023). Recent work has studied unsupervised or disentangled bottlenecks (Sawicki et al., 2024) and surveyed the challenges of concept vocabulary design, supervision, and faithfulness (Lee et al., 2024). Our work keeps the CBM decomposition but extends it from a global concept vector to spatially reusable concept components.

Spatial, object-centric, and counterfactual explanation. RoI-based architectures enable object-level prediction by using region-aligned features (Girshick, 2015; He et al., 2017). Spatial concept methods such as SALF-CBM produce concept heatmaps and allow local user edits (Benou & Riklin-Raviv, 2025); OCB associates concepts with detected instances (Steinmann et al., 2025). These methods localize evidence but do not learn target-class concept edits that satisfy the sparsity and isolation objectives. Counterfactual explanations describe minimal changes that alter a prediction (Wachter et al., 2017; Ustun et al., 2019); visual counterfactuals have been generated through region replacement, diffusion, and object-aware scene edits (Goyal et al., 2019; Augustin et al., 2022; Zemni et al., 2023). CF-CBM moves this idea into concept space but edits pooled concepts (Dominici et al., 2025). Spatial attribution and submodular selection methods identify compact regions or features (Wang et al., 2024); WHE-CBM instead learns edits whose units are reusable spatial concept components.

3. Method

3.1. Problem Setup

Each sample consists of an image x, image label y, RoIs $\mathcal{R} = \{r_i\}_{i=1}^N$, RoI labels $\boldsymbol{y}^{\text{roi}}$, foreground-validity indicators $\boldsymbol{m} = \{m_i\}_{i=1}^N$, and concept supervision at image, RoI, or pixel/grid level when available. A spatial CBM produces full-image concept logits $\boldsymbol{C}^{\text{full}} \in \mathbb{R}^{K \times H_c \times W_c}$. For a target RoI r_t, source prediction y_s, and desired target class y_t, we seek a spatial concept edit $\Delta\boldsymbol{C}$ that causes the edited representation to predict y_t while changing as few concepts and locations as possible. The editable representation is either the full concept map $\boldsymbol{C}^{\text{full}}$ or a target-RoI concept map $\boldsymbol{C}_{\text{roi}}^{(t)} \in \mathbb{R}^{K \times T \times T}$, so the intervention is localized to concept-location components rather than a single global concept vector.

3.2. Spatial Concept Supervision

For natural images without dense human concept labels, we precompute a dense concept activation map $\tilde{\boldsymbol{P}}$ using CLIP-based spatial probing (Radford et al., 2021). For each concept name, we encode prompt templates with CLIP's text encoder. We then score spatial image crops or grid cells with CLIP image-text similarity, normalize scores per concept, and store one concept channel per grid location. To derive RoI-level targets, we apply RoIAlign (He et al., 2017) to $\tilde{\boldsymbol{P}}$ for each box r_i, yielding $\boldsymbol{s}_i^{\text{roi}} \in \mathbb{R}^{K \times T \times T}$; this samples the dense concept grid inside the object region rather than recomputing concepts from a crop. The image-level target $\boldsymbol{s}^{\text{img}}$ is obtained by averaging the same activation map over a soft foreground-RoI support map, so global and RoI supervision are different aggregations of one spatial concept source. This cross-scale construction avoids training global and RoI concepts from inconsistent sources. For datasets with native concepts, such as dSprites, MNIST-Addition, and CLEVR, we use the provided concept annotations or generated concept maps directly; for CUB, we use label-free concept extraction.

3.3. WHE-CBM Architecture

Spatial bottleneck. A backbone f_θ maps the image to features $\boldsymbol{F} = f_\theta(x)$. A concept head ϕ_ψ maps features to spatial concept logits $\boldsymbol{C}^{\text{full}} = \phi_\psi(\boldsymbol{F})$. For object-level prediction, RoIAlign extracts $\boldsymbol{F}_{\text{roi}}^{(i)} = \text{RoIAlign}(\boldsymbol{F}, r_i)$, the same concept head predicts $\boldsymbol{C}_{\text{roi}}^{(i)} = \phi_\psi(\boldsymbol{F}_{\text{roi}}^{(i)})$, and average pooling gives an object concept vector $\boldsymbol{c}^{(i)}$. For image-level prediction, $\boldsymbol{C}^{\text{full}}$ is pooled into $\boldsymbol{c}^{\text{glob}}$ using RoI-support/global aggregation. A shared linear classifier $g_\omega : \mathbb{R}^K \to \mathbb{R}^L$ is applied to both pooled vectors.

Counterfactual editor. The editor G_η is class conditional and fully convolutional. Given concept logits $\boldsymbol{C}$ and target class y_t, it broadcasts a learned target-class embedding, concatenates it to $\boldsymbol{C}$, and predicts

$$\Delta\boldsymbol{C} = G_\eta(\boldsymbol{C}, y_t), \qquad \boldsymbol{C}^{\text{edit}} = \boldsymbol{C} + \alpha\Delta\boldsymbol{C}.$$

The continuous logit-space delta specifies both direction and strength. Its sign map is directly interpretable: concept k must increase or decrease at location (h, w). Magnitudes are concept-vocabulary-dependent, so we interpret absolute values primarily as within-vocabulary intervention strength.

3.4. Training

Phase 1 learns the spatial CBM:

$$\mathcal{L}_{\text{P1}} = \mathcal{L}_{\text{task}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}.$$

$\mathcal{L}_{\text{task}}$ is cross entropy over image and valid RoI labels, $\mathcal{L}_{\text{con}}$ aligns full-image, RoI, and pooled concepts to supervision,

and $\mathcal{L}_{\text{aux}}$ encourages spatial coherence, absent-concept suppression, and object-centric localization. Phase 2 freezes $f_\theta, \phi_\psi, g_\omega$ and trains only G_η:

$$\mathcal{L}_{\text{P2}} = \mathcal{L}_{\text{flip}} + \lambda_{\text{sparse}} \|\Delta \boldsymbol{C}\|_1, \qquad \mathcal{L}_{\text{flip}} = \text{CE}(\boldsymbol{z}^{\text{edit}}, y_t).$$

Here $\boldsymbol{z}^{\text{edit}}$ denotes the classifier logits after applying the edit and pooling either the target RoI concept map or the full concept map. The same editor architecture is used in both settings.

4. Experiments

Implementation details. We use CLIP ViT or ResNet backbones depending on the benchmark; VOC uses frozen pretrained visual backbones, while controlled dSprites and CLEVR runs train the lightweight spatial head and, when needed, the backbone. The concept head is $\text{Conv}_{3\times3}(d \to 256)$, ReLU, and $\text{Conv}_{1\times1}(256 \to K)$. The editor uses target embedding dimension $E = 32$ and three convolutional layers: 3×3 conv to 64 channels, 3×3 conv to 64 channels, and 1×1 conv to K, with ReLU activations. We select α and loss weights via a validation search; $\alpha = 1$ is used unless validation indicates a stronger intervention. The concept counts are $K = 7$ for dSprites, $K = 20$ for MNIST-Addition, $K = 112$ for CUB-200, $K = 15$ for CLEVR, and $K = 96$ for Pascal VOC HCS.

Datasets and baselines. We evaluate concept-space counterfactual behavior on dSprites, MNIST-Addition, and CUB-200; spatial locality on dSprites and CLEVR; and object-level behavior on Pascal VOC. Baselines include standard CBM, CF-CBM, SALF-CBM with post-hoc gradient or sparse counterfactual optimization, global concepts with RoIAlign, and OCB with sparse post-hoc counterfactual optimization.

Metrics. Validity is the fraction of counterfactuals whose prediction matches the requested target. Δ-sparsity follows CF-CBM and is reported with the same concept vocabulary. Inside-object mass (IOM) is the fraction of $|\Delta \boldsymbol{C}|$ inside the target mask; negative IOM restricts the same computation to suppressive edits. Non-target class flips and score drift measure whether an edit intended for one RoI changes nearby instances.

For concept-space counterfactuals, we evaluate all methods on the same test examples and requested target classes. A counterfactual is considered valid only if the top-1 edited prediction matches the requested target. For spatial evaluations, we upsample concept edits to the image grid and compute mask overlap against ground-truth masks.

Counterfactual validity and sparsity. Table 2 shows that WHE-CBM matches or improves CF-CBM validity while reducing Δ-sparsity under the same concept vocabulary. The gain is largest on MNIST-Addition, where the trained editor learns regularities in digit-composition changes that CF-CBM does not capture reliably.

Table 2. Counterfactual validity and Δ-sparsity (mean $\pm$ std over 3 seeds) evaluated in global concept space on dSprites, MNIST-Addition, and CUB-200. Sparsity corresponds to Δ-Sparsity as defined in CF-CBM.

Dataset	Method	Validity ↑	Δ-Sparsity ↓
dSprites	CF-CBM	100.0 ± 0.000	2.063 ± 0.116
	WHE-CBM	100.0 ± 0.000	**1.129 ± 0.481**
MNIST-Add.	CF-CBM	55.92 ± 2.56	2.13 ± 0.58
	WHE-CBM	**99.99 ± 0.00**	**0.36 ± 0.00**
CUB-200	CF-CBM	74.49 ± 0.91	16.77 ± 1.24
	WHE-CBM	**80.54 ± 0.02**	**10.22 ± 1.59**

Table 3. Spatial locality of counterfactual edits on dSprites and CLEVR. IOM is the percentage of edit magnitude inside the target object mask; Neg. IOM reports the same quantity restricted to negative edits.

	dSprites		CLEVR	
Method	IOM ↑	Neg. IOM ↑	IOM ↑	Neg. IOM ↑
SALF-CBM + Grad CF	8.25±0.80	4.73±1.67	78.02±0.76	80.08±0.45
SALF-CBM + Sparse CF	8.42±1.29	4.57±2.39	78.26±0.34	**80.53±0.28**
CF-CBM broadcast	8.46±1.27	6.01±0.94	76.92±0.13	75.29±0.49
WHE-CBM	**63.49±0.03**	**92.10±0.05**	**82.62±0.01**	79.92±0.35

Spatial locality. Table 3 shows that post-hoc and broadcast baselines can flip predictions without learning where the edit should occur. On dSprites, their edit mass is nearly background-uniform, while WHE-CBM concentrates suppressive edits on the object mask. On CLEVR, all methods benefit from cleaner object masks, but WHE-CBM still gives the strongest overall IOM.

Object-level illustration. Figure 1 summarizes the target behavior. The editor produces signed concept deltas concentrated on the target RoI: source-supporting concepts are suppressed, and target-supporting concepts are increased at object-aligned spatial regions. On Pascal VOC fixed proposals, WHE-CBM also improves object classification, reaching 97.18±0.12% RoI accuracy versus 93.59±0.03% for global concepts with RoIAlign. The same setting yields 99.24% target flips with no non-target flips and lower non-target score drift than post-hoc alternatives, supporting the qualitative isolation shown in the figure.

Ablation signal. The localization gains are not explained by sparsity alone. On dSprites, full WHE-CBM achieves 63.49% inside-object edit mass, whereas removing spatial supervision reduces the score to 9.90%, and replacing the spatial bottleneck with global concepts plus RoIAlign yields 8.50%. Removing the sparsity penalty retains reasonable flip behavior but produces fewer minimal edits. These trends support the design choice of learning spatial concept maps first and then training a shared editor over that representation, rather than applying post-hoc counterfactual optimiza-

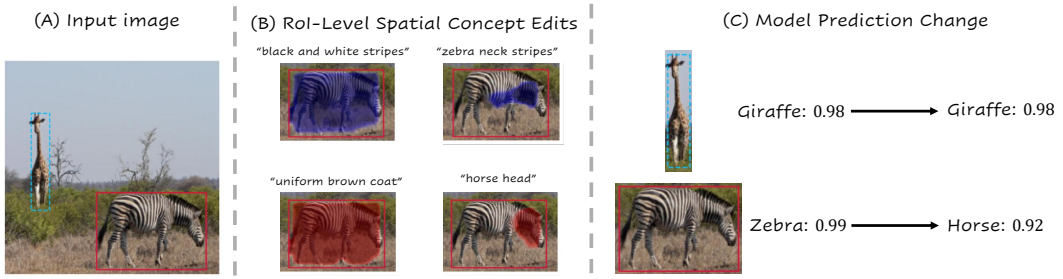

Figure 1. **Object-level counterfactual concept editing in a multi-object scene.** (A) The image contains a target RoI and a non-target RoI. (B) RoI-based editing predicts spatial concept changes, indicating where evidence should be increased or decreased. (C) Applying the edit flips the target prediction while leaving the non-target prediction unchanged.

tion to pooled concept vectors.

5. Interpretation

WHE-CBM should be read as a counterfactual explanation in the model's own concept coordinates, not as an image generator. A positive edit to "horse head" or a negative edit to "zebra neck stripes" says that, under the learned classifier, the corresponding concept evidence at those locations is what would need to change for the target decision. This is why the linear classifier and the editor play different roles. The classifier reveals how changes in concept values affect logits; the editor identifies a sparse, spatially constrained path through the concept map that achieves the requested class change.

Operationally, concepts are reusable channels, RoIAlign binds them to object instances, and the editor selects sparse concept-location edits. This decomposition allows the same vocabulary to explain multiple objects within a single image without collapsing their evidence into a single global vector. In a global CBM, "increase stripes" can only mean increasing a single scalar stripe coordinate; in WHE-CBM, it means increasing the stripe evidence at a particular spatial support for a particular RoI.

This structure gives a stronger test than attribution alone. Attribution can identify where the model looked, and CLIP-based concept attribution can attach semantic names to salient regions. WHE-CBM additionally asks an intervention question: which components of the concept would need to change to produce a specified alternative decision? Because the edit is applied within the concept bottleneck, validity can be measured directly from the modified prediction. A visually sharp edit that fails to flip the target is not a valid counterfactual; a valid edit that spreads over unrelated objects is not an object-level explanation.

The resulting claim is architectural rather than only rhetorical. Concept channels provide the reusable components, spatial location, and RoI alignment provide the binding mechanism, and the editor performs a sparse intervention over the bound components. This lets the same semantic vocabulary participate independently in different regions of the same image, which is the behavior that global concept vectors are unable to express.

6. Limitations and Conclusion

WHE-CBM depends on the quality and completeness of its concept vocabulary. Missing concepts can cause the editor to use correlated proxies, and CLIP-derived spatial supervision may inherit biases from pretrained vision-language models. Magnitudes of ΔC are meaningful within a vocabulary but should not be compared across vocabularies. Our object isolation experiments also use fixed RoIs; detector-in-the-loop counterfactuals may introduce additional interference pathways through proposal scoring, non-maximum suppression, or matching. These constraints do not invalidate the bottleneck intervention, but they define the current scope: WHE-CBM explains fixed object-level decisions in concept space. Future work should couple the bottleneck editor to proposal generation and incorporate structured concept constraints, such as attribute exclusivity, to ensure that valid edits are also more semantically plausible.

We introduced WHE-CBM, a spatial concept bottleneck model for object-level counterfactual explanation. By training a convolutional editor over RoI-aligned concept maps, WHE-CBM explains what semantic components should change and where they should change to flip a specified object prediction. The method improves concept-space validity and sparsity, spatial locality, and object-level isolation, suggesting that spatially grounded concept editing is a useful inductive bias for interpretable visual recognition. Taken together, the results separate three requirements that are often conflated: a counterfactual must flip the requested decision, remain sparse in concept space, and keep its spatial support tied to the intended object.

Acknowledgments. OL and RE were supported by the MOST grant No. 0007341.

References

Augustin, M., Boreiko, V., Croce, F., and Hein, M. Diffusion visual counterfactual explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Benou, I. and Riklin-Raviv, T. Show and tell: Visually explainable deep neural nets via spatially-aware concept bottleneck models. *arXiv preprint arXiv:2502.20134*, 2025.

Dominici, G., Barbiero, P., Giannini, F., Gjoreski, M., Marra, G., and Langheinrich, M. Counterfactual concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2025. Poster.

Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Goyal, Y., Feder, A., Shalit, U., and Kim, B. Counterfactual visual explanations. In *International Conference on Machine Learning (ICML)*, 2019.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Lee, J. H., Mikriukov, G., Schwalbe, G., Wermter, S., and Wolter, D. Concept-based explanations in computer vision. *arXiv preprint arXiv:2409.13456*, 2024.

Oikarinen, T. and Weng, T.-W. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *International Conference on Learning Representations*, 2023.

Oikarinen, T., Das, S., Nguyen, L. M., and Weng, T.-W. Label-free concept bottleneck models. In *International Conference on Learning Representations*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Sawicki, M., Bouniot, Q., Sarfraz, M. S., and Stiefelhagen, R. Discovering neurons in concept bottleneck models. *arXiv preprint arXiv:2406.00348*, 2024.

Steinmann, D., Stammer, W., Wüst, A., and Kersting, K. Object centric concept bottlenecks. *arXiv preprint arXiv:2505.24492*, 2025.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, 2019.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. In *Proceedings of the Workshop on Explainable Artificial Intelligence (XAI)*, 2017. arXiv:1711.00399.

Wang, Y., Li, Z., and Rudin, C. Less is more: Submodular attribution for structured model explanations. In *International Conference on Learning Representations*, 2024.

Yang, Y., Panagopoulou, A., Zhou, S., Jin, M., Callison-Burch, C., and Yatskar, M. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zemni, M., Chen, M., Zablocki, É., Ben-Younes, H., Pérez, P., and Cord, M. Octet: Object-aware counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

A. Supplementary Details

A.1. Losses

The Phase 1 concept objective combines binary cross entropy or mean squared error for available concept targets at image, RoI, and spatial levels:

$$\begin{aligned}\mathcal{L}_{\text{con}} = \lambda_{\text{img}} \ell(\boldsymbol{c}^{\text{glob}}, \boldsymbol{s}^{\text{img}}) \\ + \lambda_{\text{roi}} \sum_i m_i \ell(\boldsymbol{C}_{\text{roi}}^{(i)}, \boldsymbol{s}_i^{\text{roi}}) \\ + \lambda_{\text{pix}} \ell(\boldsymbol{C}^{\text{full}}, \boldsymbol{s}^{\text{pix}}).\end{aligned}$$

The auxiliary loss includes total-variation smoothing, absent-concept suppression, and object-support localization when masks are available. Phase 2 applies the same flip loss to either full-image or RoI-pooled edited concepts and uses an ℓ_1 sparsity term over $\Delta \boldsymbol{C}$.

A.2. Editor

For target class y_t, G_η computes a learned embedding $\boldsymbol{e}_t \in \mathbb{R}^{32}$, broadcasts it spatially, and predicts:

$$\begin{aligned}\Delta \boldsymbol{C} &= \text{Conv}_{1\times1}^{64\to K}\big(\text{ReLU}(\boldsymbol{H}_2)\big), \\ \boldsymbol{H}_2 &= \text{Conv}_{3\times3}^{64\to 64}\big(\text{ReLU}(\boldsymbol{H}_1)\big), \\ \boldsymbol{H}_1 &= \text{Conv}_{3\times3}^{K+32\to 64}([\boldsymbol{C}; \boldsymbol{e}_t]).\end{aligned}$$

Because this network is fully convolutional, the same editor can operate on full spatial maps and fixed-size RoI maps.