UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND IMAGE COMPUTING

Interaktive Läsionssegmentierung in der Computertomographie

Interactive Lesion Segmentation in Computed Tomography

Masterarbeit

im Rahmen des Studiengangs
Mathematik in Medizin und Lebenswissenschaften
der Universität zu Lübeck

Vorgelegt von
Constantin Ziemann

Ausgegeben und betreut von
Prof. Dr. Jan Lellmann
Institute of Mathematics and Image Computing

Mit Unterstützung von
Dr. Tanja Loßau
Fraunhofer MEVIS

17. August 2026

Eidesstattliche Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt zu haben.

Lübeck,
September 1, 2026

_____

Autor

Kurzfassung

Die Auswertung onkologischer CT-Bilder stellt Radiologen vor die Herausforderung, bei begrenzter Befundungszeit möglichst präzise Läsionsmessungen durchzuführen. Interaktive Segmentierungsmodelle bieten die Möglichkeit, bestehende Segmentierungen durch gezielte Nutzereingaben zu verfeinern. Diese Arbeit vergleicht für diese Aufgabe das Modell nnInteractive mit den in dieser Arbeit entwickelten aufgabenspezifischen Modellen. Dafür wurden drei Experimente durchgeführt, die die initiale Segmentierungsfähigkeit, das Verhalten bei iterativer Verfeinerung und die unmittelbare Reaktion auf einzelne Korrektur-Prompts untersuchten. Die aufgabenspezifischen Modelle erzielten bei der initialen Segmentierung höhere Dice-Werte als nnInteractive und führten sowohl bei der iterativen Verfeinerung als auch bei der gezielten Korrektur einzelner Fehlerkomponenten häufiger zu einer Verbesserung der bestehenden Segmentierung. Eine besondere Stärke der aufgabenspezifischen Modelle lag im Erhalt der Läsionsmaske außerhalb der zu korrigierenden Fehlerregion.

Abstract

The analysis of oncological CT images presents radiologists with the challenge of obtaining precise lesion measurements within limited reporting time. Interactive segmentation models offer the possibility of refining existing segmentations through targeted user interactions. This thesis compares the nnInteractive model with task-specific models developed in this work for this purpose. Three experiments were conducted to evaluate initial segmentation performance, behavior during interactive refinement, and the immediate response to individual correction prompts. The task-specific models achieved higher Dice scores than nnInteractive for initial segmentation and more frequently improved existing segmentations during both iterative refinement and targeted correction of individual error components. A particular strength of the task-specific models was the preservation of the lesion mask outside the error region being corrected.

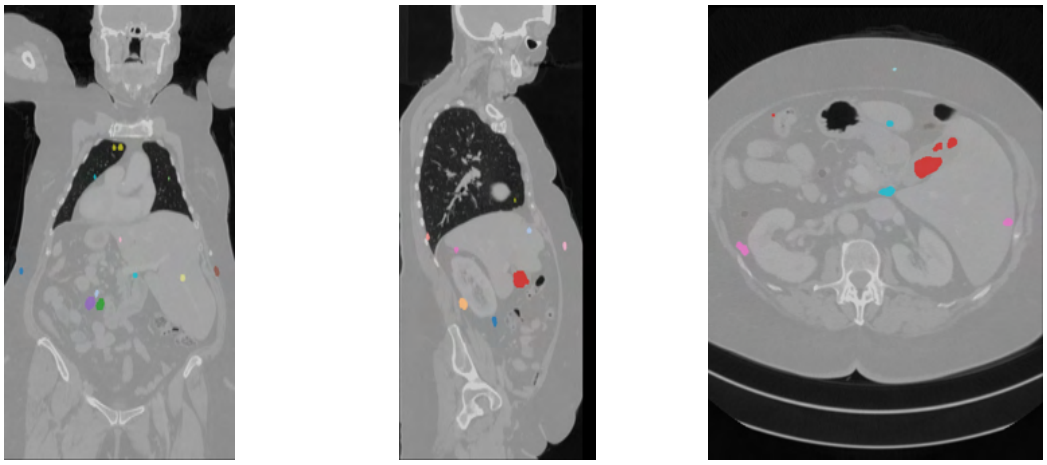
Contents

Chapter 1: Introduction

1.1 Motivation

In oncology, the examination of CT images is central to diagnosis, treatment planning, and follow-up monitoring. The clinical standard for tumor response assessment is the Response Evaluation Criteria in Solid Tumors (RECIST), which is mainly based on the sum of lesion diameters [10]. As a result, RECIST-based assessment does not explicitly account for the full three-dimensional spatial structure of lesions. Therefore, complete 3D lesion segmentation has the potential to provide additional volumetric and spatial information for tumor response assessment.

However, 3D lesion segmentation is time-consuming, which limits its integration into radiological workflows [15]. As shown in Figure 1, metastases can be spread throughout the body and often vary substantially in size and shape, making complete and precise lesion segmentation in whole-body CT images particularly challenging. Furthermore, the increasing utilization of CT imaging contributes to a growing workload for radiologists [29]. For this reason, AI-based automatic lesion segmentation has become increasingly relevant in recent years, and numerous fully automatic methods have been proposed [4, 5, 11].



(a) Coronal CT slice (b) Sagittal CT slice (c) Axial CT slice

Figure 1: **Example CT slices from a patient with melanoma.** Colored overlays indicate individual lesion segmentation masks, illustrating the variability in lesion size, shape, and anatomical location.

Nevertheless, model-generated segmentations are not sufficiently accurate in every case and therefore still require visual verification and, where necessary, manual correction [38, 27]. A single-prompt method can efficiently generate an initial segmentation of a selected lesion, but an inaccurate result must subsequently be refined. Manually correcting an almost correct three-dimensional mask using basic drawing tools would reduce

the efficiency gained through the initial model prediction. Deep-learning-based interactive refinement methods instead allow the user to indicate missing or falsely included regions through positive and negative prompts and to request an updated segmentation.

For an interactive lesion segmentation method to support a practical workflow, several aspects need to be considered. User interactions should be simple and require little effort. In this thesis, we address this by restricting refinement interactions to positive and negative point prompts. Model updates should also be sufficiently fast to allow smooth iterative refinement without interrupting the workflow. Furthermore, interactive refinement must not only correct the prompted error region but also preserve already correct parts of the existing mask, such that previously verified regions do not need to be reassessed by the user [3].

This preservation requirement is especially important when the current segmentation is already highly accurate and the remaining error regions occupy only a small part of the lesion. In such cases, unintended modifications outside the targeted region may outweigh the benefit of the intended correction. Consequently, the quality of an interactive refinement method cannot be assessed solely from its ability to generate complete segmentations. The localization and stability of individual corrections must also be considered.

General-purpose interactive segmentation models such as `nnInteractive` provide a promising basis for three-dimensional medical image segmentation and refinement [20]. `nnInteractive` has demonstrated strong general-purpose performance, including first place in the CVPR 2025 Foundation Models for Interactive 3D Biomedical Image Segmentation Challenge [2]. However, they are designed for broad applicability across different biomedical structures and interaction types rather than specifically for the refinement of individual lesion masks in CT images. It remains relevant to investigate whether prompted error regions can be corrected while already correct parts of an existing lesion segmentation are preserved.

This motivates the evaluation of `nnInteractive` on clinically motivated lesion-mask corrections and the development of task-specific refinement models. The `MultiClick` models proposed in this thesis are intended to support both initial segmentation and subsequent refinement, with particular emphasis on responding to correction prompts while preserving already correct regions of the existing mask. This thesis investigates how the proposed `MultiClick` models, trained on systematically generated lesion-refinement samples, compare with the general-purpose `nnInteractive` model for point-based lesion segmentation and refinement in CT.

1.2 Related Work

Medical segmentation in CT images has largely been driven by advances in deep learning. In particular, convolutional neural networks (CNNs) have become the standard choice [25]. One of the most influential CNN architectures for medical image segmentation is the U-Net, introduced by Ronneberger et al. in 2015 [32]. This architecture is widely used for segmentation because it combines local image details with contextual information, which is essential for accurate universal and multi-category lesion segmentation [31, 25].

Early approaches in deep-learning-based lesion segmentation commonly focused on individual organs or lesion categories. Christ et al., for example, proposed a cascaded fully convolutional approach for liver and liver-lesion segmentation [4]. More recent work has increasingly addressed universal lesion segmentation, in which lesions from different anatomical locations and categories are segmented within a common framework [31]. The Universal Lesion Segmentation 2023 Challenge provided a benchmark for three-dimensional universal lesion segmentation in CT images and demonstrated the continued relevance of task-specific segmentation pipelines based on `nnU-Net` [5, 19].

Despite these advances, fully automatic and semi-automatic predictions may remain inaccurate in challenging cases and generally require visual verification or manual correction before quantitative analysis [27]. AI-assisted workflows can reduce this manual effort by combining automatic lesion segmentation with user input. Hering et al. investigated such an AI-assisted workflow for longitudinal CT assessment and showed the potential of registration and volumetric segmentation assistance in a reader study [15].

A particularly relevant example for this thesis is the MEVIS-internal `OneClick` Lesion Segmentation tool. After the user places a single point inside the lesion of interest, an `nnU-Net`-based model predicts a three-dimensional lesion mask from a local CT image patch [5]. This approach provides an efficient way to generate an initial lesion segmentation. However, its interaction is limited to the initial point prompt, and the method does not provide a mechanism for further interactive refinement if the resulting mask is inaccurate.

Interactive medical image segmentation addresses this limitation by incorporating user feedback into the segmentation process. Depending on the method, users can provide positive or negative point prompts, scribbles, bounding boxes, or other spatial annotations. Marinov et al. distinguished interactive segmentation methods according to, among other aspects, the type of interaction, the stage at which the interaction is incorporated, and the way repeated interactions are used to refine a segmentation [27]. Sakinis et al. proposed a click-based interactive segmentation approach in which positive and negative user interactions guide the segmentation of medical images [34]. Building on this concept, `DeepEdit` combines automatic volumetric segmentation with subsequent click-based interactive refinement within a single model [6].

Several interactive segmentation approaches explicitly incorporate an existing segmentation into subsequent correction steps. `DeepIGeoS` combines an initial segmentation with user-provided foreground and background interactions to refine segmentation errors [38]. Previous-mask guidance has also been used in iterative click-based general 2D segmentation approaches, where the output of a preceding interaction can be provided to the model during subsequent corrections [36]. These approaches demonstrate that an existing segmentation can serve as important contextual information during interactive refinement.

Beyond incorporating the previous mask, other methods have addressed the locality of interactive corrections. `FocalClick`, for example, was motivated by the observation that updating an entire segmentation in response to local interactions can unintentionally modify already correct regions. It therefore performs localized mask correction and

preserves regions outside the selected refinement area [3]. This highlights an important distinction between generating a complete segmentation and refining an already accurate mask: successful refinement requires both responsiveness to the new interaction and preservation of unaffected regions.

More recently, promptable segmentation models have aimed to generalize across a wider range of structures and segmentation tasks. The Segment Anything Model introduced a general prompt-based segmentation framework trained on a large collection of natural two-dimensional images [22]. `MedSAM` adapted this concept to medical image segmentation [26], while `SAM-Med3D` extended promptable segmentation to volumetric medical image data [39]. These developments demonstrate the potential of general prompt-based models, but differences between natural and medical images and limited volumetric context in two-dimensional approaches remain important challenges. `SegVol` further extended this direction toward universal interactive volumetric segmentation in CT, supporting spatial and semantic prompts for a broad range of anatomical structures [8]. `VISTA3D` similarly combines automatic and interactive three-dimensional medical image segmentation within a unified model and explicitly targets the efficient correction of automatic predictions [14].

`nnInteractive` was introduced as a general-purpose promptable segmentation model for three-dimensional biomedical images [20]. The method builds on the `nnU-Net` framework and supports several interaction types, including positive and negative points, scribbles, bounding boxes, and lassos. It receives the medical image, the current segmentation state, and the prompt information and can be applied both to initial segmentation and iterative refinement. While `SAM-Med3D` follows the `SAM` paradigm with separate three-dimensional image, prompt, and mask encoders and primarily relies on point prompts, `nnInteractive` incorporates spatial prompts directly as additional input channels to an `nnU-Net`-based segmentation network and supports a broader range of corrective interaction types. In contrast to methods developed for a single organ or image modality, `nnInteractive` was trained on heterogeneous biomedical datasets and was designed for broad applicability.

This thesis complements this general-purpose approach by investigating task-specific models for interactive lesion segmentation in CT. Rather than targeting a broad range of biomedical segmentation tasks and interaction types, the proposed `MultiClick` models are trained specifically on lesion-segmentation and refinement samples using only point prompts.

1.3 Contribution and Outline

This thesis investigates point-based interactive segmentation and refinement of volumetric lesions in CT images, with a focus on the correction of local segmentation errors while preserving already correct regions of an existing mask. To this end, the general-purpose `nnInteractive` model is compared with task-specific `MultiClick` models developed in this work.

The first contribution is a systematic evaluation setting for interactive segmentation and refinement. In addition to conventional global segmentation metrics, the evaluation

considers the response to individual correction prompts, the preservation of mask regions outside the targeted error component, and characteristics of the underlying segmentation errors. This allows the models to be assessed not only by the quality of their final segmentation, but also by how locally and reliably they modify an existing mask.

The second contribution is the development of a controlled training-data generation strategy for point-based lesion refinement. Starting from imperfect `OneClick` segmentations, interaction sequences are constructed from segmentation-error components. Two training-data choices are investigated: whether the training target was a stepwise correction or the complete reference segmentation, and whether small error-component samples were retained or removed from the training dataset. Their combination results in four task-specific model variants.

The third contribution is a comparative experimental evaluation of `nnInteractive` and the four `MultiClick` models. The models are evaluated on a MEVIS-internal dataset containing real radiological corrections, which we additionally characterize statistically. Three complementary setups test initial lesion segmentation, iterative five-click refinement, and isolated error-component correction. Together, they examine the trade-off between correction responsiveness, segmentation quality, and preservation of non-targeted mask regions.

The remainder of this thesis is organized as follows. Chapter 2 introduces the mathematical and technical concepts required for the subsequent data-generation and evaluation procedure. Chapter 3 presents `nnU-Net`, the `OneClick` Lesion Segmentation model, and `nnInteractive` as the basis on which this work builds. Chapter 4 describes and analyzes the evaluation and training datasets, with particular emphasis on lesion characteristics and connected segmentation-error components. Based on these observations, Chapter 5 introduces the training-data generation procedure and the four proposed `MultiClick` model variants. Chapter 6 presents the experimental evaluation and compares the models across the three interaction settings. Finally, Chapter 7 discusses the main findings, limitations, and directions for future work.

Chapter 2: Foundations

Interactive lesion refinement in this thesis depends on spatial relationships between a current segmentation, a reference mask, and point prompts defined on a three-dimensional voxel grid. Sections 2.1, 2.2, 2.3, and 2.4 therefore establish the notation required to describe segmentation masks, connected error components, point-prompt locations, and transformations between voxel and physical image space. The segmentation models considered in the following chapters are based on three-dimensional U-Net architectures within the `nnU-Net` framework. The final sections 2.5 and 2.6 of this chapter briefly introduce the CNN and U-Net concepts required to describe their network architecture and training objective.

The representation of images as spatial objects with an associated voxel grid, spacing, origin, and direction follows the conventions used in SimpleITK [35]. Similar set-based representations of binary segmentation volumes have been used in medical image analysis [23, 41].

2.1 CT Images and Segmentation Masks

Let $n_x, n_y, n_z \in \mathbb{N}$ denote the number of voxels along the three image dimensions. The finite discrete voxel grid is defined as

$$\Omega = \{0, \dots, n_x - 1\} \times \{0, \dots, n_y - 1\} \times \{0, \dots, n_z - 1\} \subseteq \mathbb{Z}^3.$$

A **computed tomography image** (CT image) is modeled as a function

$$\begin{aligned} I : \Omega &\to \mathbb{R}, \\ \omega &\mapsto I(\omega), \end{aligned}$$

where every voxel position $\omega \in \Omega$ is assigned an intensity value $I(\omega) \in \mathbb{R}$. A corresponding binary **segmentation mask** M is defined on the same voxel grid Ω as a function

$$\begin{aligned} M : \Omega &\to \{0, 1\}, \\ \omega &\mapsto M(\omega). \end{aligned}$$

For each voxel position $\omega \in \Omega$, the value $M(\omega)$ indicates whether the voxel belongs to the target structure. In the case of lesion segmentation,

$$M(\omega) = \begin{cases} 1, & \text{if } \omega \text{ belongs to a lesion,} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, the **empty mask** is given by

$$\mathbf{0}_\Omega : \Omega \to \{0, 1\}, \qquad \mathbf{0}_\Omega(\omega) := 0 \qquad \forall \omega \in \Omega.$$

For a binary mask $M : \Omega \to \{0, 1\}$, we define its **foreground voxel set** as

$$\mathcal{M} := \mathrm{fg}(M) = \{\omega \in \Omega \mid M(\omega) = 1\}.$$

Conversely, for a voxel set $\mathcal{M} \subseteq \Omega$, the corresponding binary indicator mask is defined as

$$\mathbb{1}_{\mathcal{M}} : \Omega \to \{0, 1\}, \qquad \mathbb{1}_{\mathcal{M}}(\omega) = \begin{cases} 1, & \omega \in \mathcal{M}, \\ 0, & \omega \in \Omega \setminus \mathcal{M}. \end{cases}$$

The mask size is given by the cardinality of its foreground voxels:

$$|\mathcal{M}| = \sum_{\omega \in \Omega} \mathbb{1}_{\mathcal{M}}(\omega).$$

In the following, set operations are applied to foreground voxel sets. A binary mask and its foreground set are distinguished by using **Roman letters for mask functions** and **calligraphic letters for voxel sets**.

While Ω describes a discrete voxel grid, it does not encode the physical image geometry. The physical distances between neighboring voxel centers along the three image axes are given by the **voxel spacing**

$$s = (s_x, s_y, s_z) \in \mathbb{R}^3_{>0}, \tag{1}$$

where s_x, s_y and s_z denote the physical distances between the neighboring voxel centers along the three spatial axes, measured in millimeters. With this, we can describe the physical volume by $V(\mathcal{M}) = |\mathcal{M}| s_x s_y s_z$ in mm^3.

From the voxel spacing, the physical position x of a voxel ω can be obtained by scaling the voxel index component-wise and adding the image origin. This mapping can be written as

$$x : \Omega \to \mathbb{R}^3, \qquad x(\omega) = x_{\mathrm{org}} + D \,\mathrm{diag}(s)\omega, \tag{2}$$

where $x_{\mathrm{org}} \in \mathbb{R}^3$ denotes the image origin and D denotes the orthogonal direction matrix, encoding the direction of the image axes in physical space.

Furthermore, we define a **slice** as a two-dimensional cross-section of the image obtained by fixing one coordinate of the voxel grid. For example, for $k \in \{0, \ldots, n_z - 1\}$ we have the slice

$$I_k(i, j) = I(i, j, k)$$

for all $(i, j) \in \{0, \ldots, n_x - 1\} \times \{0, \ldots, n_y - 1\}$. The CT volumes used in this thesis consist of axial slices ordered along the cranio-caudal direction. Throughout this thesis, the third voxel-grid coordinate is used to index these axial slices.

2.2 Distances and Maximally Interior Voxel Selection

To clearly distinguish between voxel-grid coordinates and physical-space coordinates, we introduce two different distance definitions. For two voxel coordinates $p, q \in \Omega$, the Euclidean distance between two voxel-grid positions is denoted by

$$d_{\mathrm{grid}}(p, q) = \|p - q\|_2,$$

and the physical Euclidean distance is denoted by

$$d_{\mathrm{phys}}(p, q) = \|x(p) - x(q)\|_2.$$

The value of d_{phys} gives us the physical distance between two voxels in millimeters. For a voxel $p \in \Omega$ and a voxel set $\mathcal{S} \subseteq \Omega$ with $\mathcal{S} \neq \emptyset$, the distance from p to $\mathcal{S}$ is defined as

$$d_{\text{grid}}(p, \mathcal{S}) = \min_{q \in \mathcal{S}} d_{\text{grid}}(p, q).$$

Thus, $d_{\text{grid}}(p, \mathcal{S})$ denotes the distance from p to the closest voxel in $\mathcal{S}$. Analogously, for a voxel $p \in \Omega$ and a voxel set $\mathcal{S} \subseteq \Omega$ with $\mathcal{S} \neq \emptyset$, the physical distance from p to $\mathcal{S}$ is defined as

$$d_{\text{phys}}(p, \mathcal{S}) = \min_{q \in \mathcal{S}} d_{\text{phys}}(p, q). \tag{3}$$

For a voxel set $\mathcal{C} \subseteq \Omega$ with $\mathcal{C} \neq \emptyset$, the **Euclidean distance transform (EDT)** within $\mathcal{C}$ assigns to each foreground voxel its distance to the complement $\Omega \setminus \mathcal{C}$:

$$D_{\mathcal{C}}(p) = d_{\text{grid}}(p, \Omega \setminus \mathcal{C}) = \min_{q \in \Omega \setminus \mathcal{C}} \|p - q\|_2, \qquad p \in \mathcal{C}.$$

Thus, $D_{\mathcal{C}}(p)$ measures how far a voxel p lies from the background of $\mathcal{C}$. Voxels close to the boundary of $\mathcal{C}$ have small distance-transform values, whereas more central voxels have larger values.

A maximal interior voxel $p_{\mathcal{C}}$ can then be selected as a voxel that attains the maximum distance-transform value,

$$p_{\mathcal{C}} \in \arg\max_{p \in \mathcal{C}} D_{\mathcal{C}}(p). \tag{4}$$

Such a voxel is maximally distant from the background $\Omega \setminus \mathcal{C}$ and can be interpreted as the *center voxel of the largest inscribed sphere (CoLIS)* in the discrete-grid sense. Since all computations were performed on the discrete voxel grid, the selected point is restricted to voxel locations and distances are measured in voxel-grid units. The maximizer does not have to be unique. The specific distance-transform implementation and dimensionality used for this point prompt generation are described in the corresponding data-generation and evaluation sections.

2.3 Connected Error Components

In this thesis, we distinguish between the initial segmentation mask A, usually generated by a model, and the reference mask R, which serves as the target segmentation for interactive lesion segmentation models. Both masks are binary masks defined on the same finite voxel grid. Their corresponding foreground voxel sets are denoted by $\mathcal{A} = \text{fg}(A)$ and $\mathcal{R} = \text{fg}(R)$. Regions that are present in the reference mask but missing in the automatic mask correspond to the set of *include regions*,

$$\mathcal{E}^{+} = \mathcal{R} \setminus \mathcal{A} = \{\omega \in \Omega \mid R(\omega) = 1 \wedge A(\omega) = 0\},$$

whereas regions that are present in the automatic mask but absent from the reference mask correspond to the set of *exclude regions*,

$$\mathcal{E}^{-} = \mathcal{A} \setminus \mathcal{R} = \{\omega \in \Omega \mid A(\omega) = 1 \wedge R(\omega) = 0\}.$$

These terms are used because the corresponding regions must either be included in or excluded from the initial automatic segmentation mask.

The resulting error regions are further decomposed into connected error components. The use of neighborhoods to define connectivity is a standard concept in connected component labeling and is further described in Rosenfeld et al. [33]. For a point $v = (v_1, v_2, v_3) \in \mathbb{R}^3$, the maximum norm is defined as

$$\|v\|_\infty := \max_{i \in \{1,2,3\}} |v_i|.$$

For a voxel $\omega \in \Omega$, its **26-neighborhood** is defined as

$$\mathcal{N}_{26}(\omega) = \{\tilde{\omega} \in \Omega \mid \|\tilde{\omega} - \omega\|_\infty = 1\}.$$

This means that for a voxel $\tilde{\omega} \in \Omega$ with $\tilde{\omega} \neq \omega$,

$$\tilde{\omega} \in \mathcal{N}_{26}(\omega) \Leftrightarrow \omega \text{ and } \tilde{\omega} \text{ share a face, an edge, or a corner.}$$

A voxel set $\mathcal{C} \subseteq \Omega$ with $\mathcal{C} \neq \emptyset$ is called **26-connected** if, for every pair $\omega, \tilde{\omega} \in \mathcal{C}$, there exists $m \in \mathbb{N}_0$ and a finite sequence

$$(\omega_0, \omega_1, \ldots, \omega_m)$$

of voxels in $\mathcal{C}$ such that

$$\omega_0 = \omega, \qquad \omega_m = \tilde{\omega},$$

and

$$\omega_{j+1} \in \mathcal{N}_{26}(\omega_j) \qquad \text{for all } j \in \{0, \ldots, m-1\}.$$

The case $m = 0$ ensures that a voxel set containing a single voxel is considered 26-connected. A subset $\mathcal{C} \subseteq \mathcal{S} \subseteq \Omega$ is called a **26-connected component** of $\mathcal{S}$ if

1. $\mathcal{C}$ is 26-connected, and
2. $\mathcal{C}$ is maximal with respect to set inclusion, that is, there exists no 26-connected set $\tilde{\mathcal{C}} \subseteq \mathcal{S}$ such that $\mathcal{C} \subset \tilde{\mathcal{C}}$.

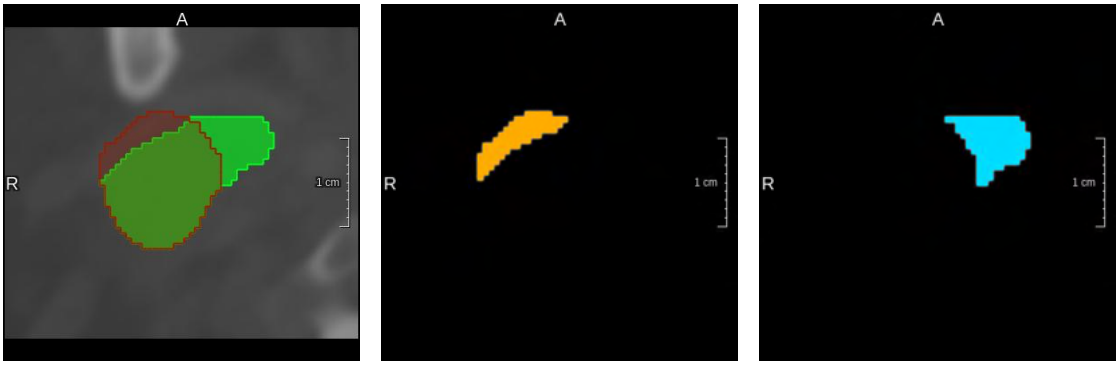
The maximality condition means that $\mathcal{C}$ cannot be enlarged within $\mathcal{S}$ while remaining 26-connected.

The **set of all 26-connected components** of $\mathcal{S} \subseteq \Omega$ is denoted by

$$\mathcal{K}_{26}(\mathcal{S}) := \{\mathcal{C} \subseteq \mathcal{S} \mid \mathcal{C} \text{ is a 26-connected component of } \mathcal{S}\}.$$

From this, we retrieve the sets $\mathcal{K}_{26}(\mathcal{E}^+)$ and $\mathcal{K}_{26}(\mathcal{E}^-)$ containing the include and exclude error components, respectively. The elements of $\mathcal{K}_{26}(\mathcal{E}^+)$ are called **include error components**, whereas the elements of $\mathcal{K}_{26}(\mathcal{E}^-)$ are called **exclude error components**.

An example of include and exclude error components derived from the comparison of an automatic lesion segmentation and its manually corrected version is illustrated in Figure 2. This figure shows one axial cross-section of the three-dimensional masks and error components.



(a) Automatic and reference segmentation masks. (b) Exclude error component. (c) Include error component.

Figure 2: **Example of connected error regions derived from an automatic lesion segmentation (red) and its corresponding reference mask (green).** In (a), the automatic and reference masks are superimposed on the axial CT image slice. The exclude error component in (b) contains voxels that are present in the automatic mask but absent from the reference mask. The include error component in (c) contains voxels that are present in the reference mask but missing from the automatic mask. This example is a two-dimensional cross-section of a three-dimensional lesion mask.

2.4 Anisotropy, Coordinate Transformations, and Resampling

An image is called **anisotropic** if the voxel spacing differs between spatial directions. For a spacing vector $s = (s_x, s_y, s_z) \in \mathbb{R}^3_{>0}$, this is the case if

$$\frac{\max\{s_x, s_y, s_z\}}{\min\{s_x, s_y, s_z\}} > 1.$$

In the volumetric CT images used in this thesis, the in-plane resolution is typically higher than the through-plane resolution. Consequently, a one-voxel displacement along the slice direction may correspond to a larger physical distance than a one-voxel displacement within an axial slice. Distances, object sizes, and anatomical structures represented in voxel units therefore depend on the image resolution, which can make it more difficult for neural networks to learn consistent shape and scale patterns [40].

To reduce spacing variability, an image can be resampled from its original spacing

$$s = (s_x, s_y, s_z)$$

to a target spacing

$$\tilde{s} = (\tilde{s}_x, \tilde{s}_y, \tilde{s}_z).$$

Let $\tilde{\Omega}$ denote the finite target voxel grid and let $\tilde{x} : \tilde{\Omega} \to \mathbb{R}^3$ denote its mapping to physical space. For a target grid position $\tilde{\omega} \in \tilde{\Omega}$, the corresponding continuous coordinate in the original voxel grid is

$$T(\tilde{\omega}) := \operatorname{diag}(s)^{-1} D^{-1} \left(\tilde{x}(\tilde{\omega}) - x_{\text{org}}\right).$$

This transformation is well defined because all components of the voxel spacing s are strictly positive, such that $\mathrm{diag}(s)$ is invertible, and because the direction matrix D is orthogonal, such that $D^{-1} = D^{\top}$. Since $T(\tilde{\omega})$ does not generally coincide with an exact voxel position, interpolation is required to estimate the image intensity at this intermediate position. Let $I_{\mathrm{inter}} : \mathbb{R}^3 \to \mathbb{R}$ denote the interpolant used to estimate image intensities at non-integer voxel-grid coordinates. The resampled image is defined as

$$\tilde{I}(\tilde{\omega}) := I_{\mathrm{inter}}(T(\tilde{\omega})).$$

Changing the voxel spacing generally also changes the number of voxels. To approximately preserve the distance between the first and the last voxel centers along each image axis, the original and target grid dimensions satisfy

$$(\tilde{n}_i - 1)\tilde{s}_i \approx (n_i - 1)s_i.$$

Since the target grid dimension $\tilde{n}_i$ must be an integer, the original physical extent cannot in general be reproduced exactly for an arbitrary target spacing. For example, resampling from a slice spacing of $s_z = 2.5\,\mathrm{mm}$ to a target spacing of $\tilde{s}_z = 1\,\mathrm{mm}$ increases the sampling resolution in the z-direction by a factor of 2.5:

$$\tilde{n}_z - 1 \approx 2.5(n_z - 1).$$

CT images contain scalar intensity values and can therefore be resampled using interpolation methods that estimate intermediate intensity values. Binary segmentation masks instead represent categorical labels. Consequently, nearest-neighbor interpolation is used for masks to preserve the discrete range $\{0, 1\}$:

$$\tilde{M}(\tilde{\omega}) = M\left(\mathrm{round}(T(\tilde{\omega}))\right),$$

where the rounding operation is applied component-wise.

For CT intensities, a higher-order interpolation method used later in this thesis for training data generation and model inference is the Lanczos interpolation with a window size of $a = 3$. Its one-dimensional kernel is

$$L_3(u) = \begin{cases} \mathrm{sinc}(u)\,\mathrm{sinc}\left(\frac{u}{3}\right), & |u| < 3, \\ 0, & |u| \geq 3, \end{cases}$$

where

$$\mathrm{sinc}(u) = \begin{cases} \dfrac{\sin(\pi u)}{\pi u}, & u \neq 0, \\ 1, & u = 0. \end{cases}$$

For multidimensional interpolation, the kernel can be applied separably along the resampled image axes. For example, at the continuous position $z' = 10.4$, the Lanczos-3 interpolation uses the neighboring slices $k \in \{8, 9, 10, 11, 12, 13\}$. The interpolation can be represented schematically as

$$I_{\mathrm{L3}}(i, j, 10.4) = \sum_{k=8}^{13} I(i, j, k)\, L_3(10.4 - k).$$

For interpolation along the slice direction, the in-plane coordinates remain fixed. The new intensity is calculated as a weighted combination of the six neighboring slice intensities.

Lanczos interpolation provides smooth intensity estimates while retaining image detail and is therefore applicable to continuously-valued CT intensities. For further information on Lanczos interpolation and windowed sinc filters, see [9].

2.5 Three-Dimensional Convolutional Neural Networks

Convolutional neural networks (CNNs) are widely used for the analysis of medical images because they can learn spatial image features directly from training data [25]. A three-dimensional CNN processes a volumetric input

$$X \in \mathbb{R}^{C_{\text{in}} \times n_x \times n_y \times n_z},$$

where C_{in} denotes the number of input channels and n_x, n_y, n_z denote spatial dimensions.

A **convolutional layer** applies a set of trainable kernels to local voxel neighborhoods. For an output feature channel c, this operation can be written schematically as

$$Y_c(p) = \sigma\left(b_c + \sum_{c'=1}^{C_{\text{in}}} \sum_{q \in \mathcal{N}} K_{c,c'}(q)\, X_{c'}(p+q)\right),$$

where p denotes a voxel position, $\mathcal{N}$ is the local kernel neighborhood, $K_{c,c'}$ contains the trainable kernel weights, b_c is a bias term, and σ denotes a nonlinear activation function. The same kernel weights are applied at every spatial position. This weight sharing allows the network to detect similar image patterns independently of their location. Strictly speaking, the operation above corresponds to a filter operation rather than a mathematical convolution. However, this operation is referred to as a convolution in deep-learning frameworks and in the literature.

Convolutional layers are commonly combined with normalization and nonlinear activation functions. Normalization stabilizes the distribution of intermediate feature representations, whereas nonlinear activation functions enable the network to represent nonlinear mappings. The networks used in this thesis apply instance normalization followed by a leaky rectified linear unit (LeakyReLU). The latter is defined as

$$\text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0, \\ \alpha x, & x < 0, \end{cases}$$

where $\alpha > 0$ determines the slope for negative inputs. In contrast to the standard ReLU, LeakyReLU therefore retains a small non-zero gradient for negative activation values.

By stacking multiple convolutional layers, the network learns increasingly complex representations. Early layers typically respond to local intensity changes and boundaries, whereas deeper layers combine information from larger spatial regions. In three-dimensional CNNs, the kernels extend across all three spatial dimensions and can therefore incorporate information from neighboring CT slices [42]. Compared with two-dimensional networks, three-dimensional networks provide additional volumetric context but also increase memory and computational requirements.

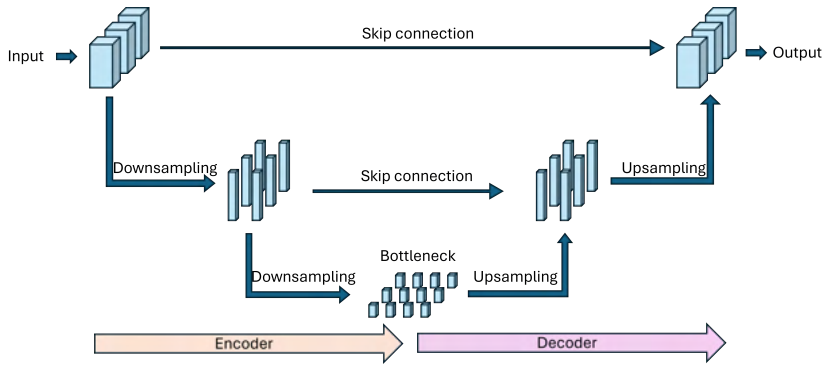


Figure 3: **Simplified schematic representation of the U-Net architecture.** The encoder progressively reduces the spatial resolution of the feature maps while increasing the number of feature channels. After the bottleneck, the decoder restores the original spatial resolution while reducing the number of feature channels. Skip connections transfer high-resolution feature information from the encoder to the corresponding decoder stage.

2.6 U-Net Architecture and Segmentation Loss

The U-Net is an encoder–decoder CNN architecture originally proposed for biomedical image segmentation by Ronneberger et al. [32]. Its encoder gradually reduces the spatial resolution of feature maps while increasing the number of feature channels. This enables the network to extract contextual information from progressively larger image regions.

The decoder subsequently restores the spatial resolution and produces a voxel-wise segmentation prediction. Skip connections transfer feature maps from encoder to decoder stages at the corresponding spatial resolution. These connections combine the contextual information from deeper layers with spatial details from earlier layers, which supports the localization of object boundaries.

The three-dimensional U-Net extends this concept to volumetric images by replacing two-dimensional convolution, downsampling, and upsampling operations with their three-dimensional counterparts [42]. For binary lesion segmentation, the final network layer produces a foreground probability

$$\hat{y}(p) \in [0, 1]$$

for every voxel position p in the output region. A binary segmentation mask can then be obtained by applying a probability threshold. Figure 3 shows a schematic illustration of the U-Net architecture.

During supervised training, the prediction probabilities are compared with a binary target mask. A commonly used objective combines voxel-wise cross-entropy with a soft Dice loss. For predicted foreground probabilities $\hat{y}_i$ and target labels $y_i \in \{0, 1\}$, the soft Dice loss can be defined as

$$L_{\text{Dice}} = 1 - \frac{2 \sum_i \hat{y}_i y_i + \varepsilon}{\sum_i \hat{y}_i + \sum_i y_i + \varepsilon},$$

where $\varepsilon = 10^{-5}$ is a small smoothing constant used for numerical stability. This value was inherited from the standard `nnU-Net` training configuration and was not tuned separately in this thesis. The Dice loss directly optimizes the overlap between prediction and target and is therefore useful for segmentation tasks with strongly imbalanced foreground and background classes. Cross-entropy provides an additional voxel-wise classification objective. Their combination is commonly used in medical image segmentation and is also employed by the `nnU-Net` training framework [19] described in Section 3.1.

Chapter 3: Segmentation Frameworks and User-Assisted Methods

This chapter introduces the segmentation frameworks that form the methodological basis of this thesis. First, `nnU-Net` is presented as a general self-configuring framework for biomedical image segmentation and as the training framework used for the proposed MultiClick models. Subsequently, the MEVIS-internal `OneClick` Lesion Segmentation model is described, which provides the initial lesion segmentations used throughout this work for training-data generation. Finally, `nnInteractive` is introduced as the general-purpose interactive segmentation baseline used in the experimental evaluation.

3.1 nnU-Net

The performance of deep learning-based segmentation methods strongly depends on task-specific design choices, such as preprocessing, patch size, network topology, training schedule, and postprocessing [25]. Selecting appropriate configurations for these design choices requires substantial expertise, and even slightly suboptimal choices can lead to a considerable drop in segmentation performance. The `nnU-Net` framework [19] was introduced to address the challenge of making appropriate design choices. It provides a self-configuring training pipeline, thereby establishing a strong baseline for biomedical segmentation tasks [5].

Dataset fingerprint and configuration planning.
Instead of proposing a fundamentally new network architecture, `nnU-Net` provides a standardized segmentation pipeline that configures preprocessing, network architecture, training, and postprocessing based on dataset-specific properties. The design decisions made in the pipeline are separated into three categories:

1. **Fixed** parameters
2. **Rule-based** parameters
3. **Empirical** parameters

Fixed parameters are design choices that do not need to be adapted across different datasets, as they were found to work robustly for a wide range of biomedical segmentation tasks. Examples include the learning rate, loss function, and data augmentation methods. Rule-based parameters are automatically derived from a dataset fingerprint, which captures relevant properties such as image size, voxel spacing, modality, and number of classes. These parameters include, for example, decisions related to resampling, intensity normalization, patch size, and network topology. Empirical parameters are selected based on experimental evaluation, such as the choice of the best-performing model configuration or postprocessing strategy.

Based on the dataset fingerprint, `nnU-Net` generates a set of *plans* specifying the resulting preprocessing and network configuration. These plans include, among other

properties, the target voxel spacing, patch and batch sizes, network topology, and intensity normalization strategy. Depending on the dataset properties, `nnU-Net` provides different configurations, including two-dimensional and three-dimensional models. The three-dimensional full-resolution configuration processes three-dimensional patches at the planned target resolution and is the configuration used for the proposed MultiClick models in this thesis.

Network input and patch-based training.
For three-dimensional configurations, training is performed on spatial image patches rather than necessarily processing an entire volume at once. The patch size is determined as part of the `nnU-Net` configuration while accounting for the image geometry and available computational resources. During training, the network predicts a segmentation for the corresponding patch.

The network input is not restricted to a single image channel. Multiple spatially aligned input channels can be provided jointly and are concatenated along the channel dimension before entering the network. Their number determines the number of input feature channels of the first convolutional layer. This property is later used for the proposed MultiClick models to jointly provide the CT image, the current segmentation mask, and the positive and negative prompt representations.

Training objective.
During training, `nnU-Net` applies a compound segmentation objective combining Dice and cross-entropy losses. In addition, deep supervision is used by generating auxiliary segmentation outputs at intermediate decoder resolutions. The corresponding auxiliary losses provide training signals at several spatial resolutions, while only the final full-resolution prediction is required during inference.

The self-configuring design makes `nnU-Net` suitable as a general framework for developing task-specific medical image segmentation models. The `OneClick` Lesion Segmentation model and `nnInteractive`, described in the following sections, are both based on `nnU-Net`. In addition, the proposed MultiClick models developed in this thesis are trained directly using the `nnU-Net` framework.

In its standard formulation, `nnU-Net` is designed for automatic segmentation and does not define how user interactions are incorporated during inference. User-assisted segmentation methods can extend this framework by introducing spatial information that specifies the structure or region of interest [15, 27]. The following sections present two such approaches relevant to this thesis: the `OneClick` Lesion Segmentation model, in which a user-selected point determines the local image region to be segmented, and `nnInteractive`, which explicitly incorporates user interactions as additional spatial input channels.

3.2 OneClick Lesion Segmentation Model

The MEVIS-internal `OneClick` Lesion Segmentation tool is a user-assisted, patch-based method for three-dimensional lesion segmentation. The approach was assessed in the

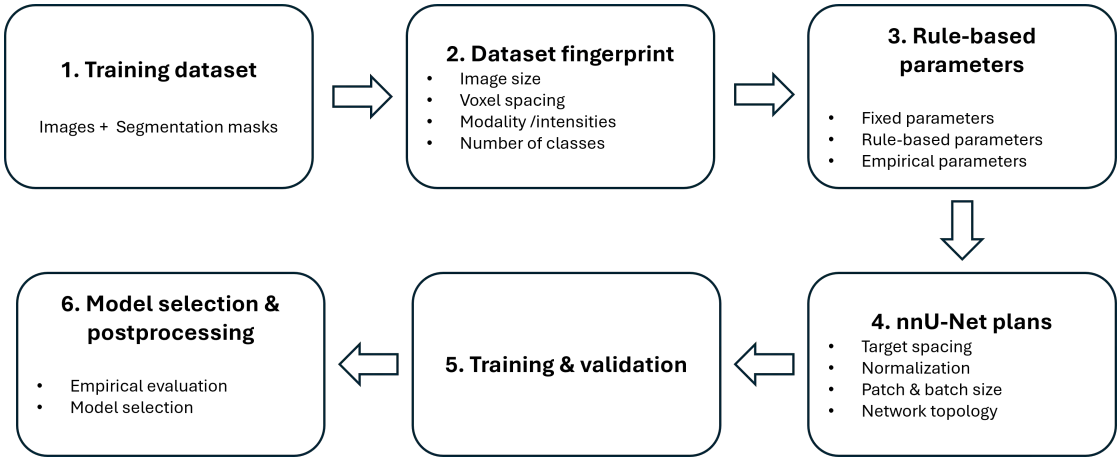


Figure 4: **Simplified schematic overview of the `nnU-Net` self-configuring segmentation pipeline.** Dataset-specific properties are summarized in a dataset fingerprint and used to derive preprocessing and network configurations. The resulting configurations are trained and validated, followed by empirical model selection and postprocessing. Based on Isensee et al. [19].

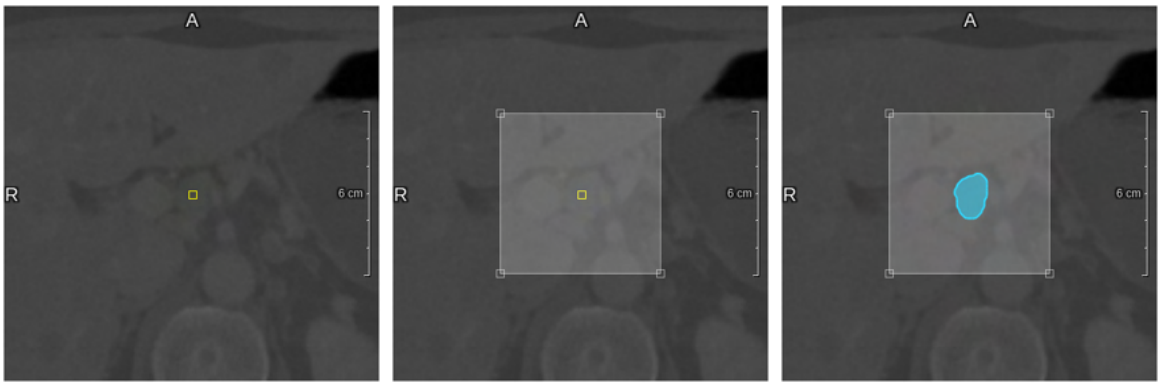
Universal Lesion Segmentation '23 Challenge [5]. The evaluation dataset used in this thesis contains initial lesion masks generated with the `OneClick` tool, together with the corresponding masks obtained after radiological review. In addition, the tool was applied to the public Longitudinal-CT data [24] to generate the initial masks required for constructing the refinement data for the proposed models.

The `OneClick` Lesion Segmentation model was developed from a patch-based segmentation concept similar to the baseline-segmentation stage described in Hering et al. [16]. In contrast to the complete longitudinal pipeline proposed in that work, the `OneClick` tool does not include the subsequent registration and follow-up lesion-tracking stages.

The approach is designed to integrate well into an AI-based interactive radiological workflow, in which the user places a single point inside the lesion of interest and receives a complete three-dimensional lesion mask. This workflow is depicted in Figure 5. The selected point determines the center of a local CT image patch that is passed to an `nnU-Net`-based segmentation model. The point is not represented through a separate input channel but is encoded implicitly through its position at the center of the extracted patch.

The implementation used in this thesis receives a cropped CT image of size $64 \times 64 \times 64$ voxels and predicts a binary mask of the same size. This differs from the ULS23 implementation described by Moltz et al., which used processing patches of size $128 \times 128 \times 128$ voxels [5, 30].

Training.
The published ULS23 model was trained on a combination of internal and publicly available datasets covering bone, liver, lung, lymph-node, soft-tissue, and other lesion



(a) Initial click point.

(b) Crop patch around the click point and apply the model.

(c) Model output mask.

Figure 5: **Procedure of the `OneClick` Lesion Segmentation tool.** A $64 \times 64 \times 64$ voxel patch is cropped from the CT image, with the user-selected click point at its center. The cropped patch is then passed to an `nnU-Net`-based model, which predicts a 3D lesion mask.

types. For each annotated lesion, one training patch was extracted. Its center was initialized at the lesion center of mass and randomly displaced by up to 32 voxels along each spatial axis. This procedure was intended to reduce the sensitivity of the model to imprecise point placement [5, 30].

Preprocessing, spatial resampling, and postprocessing.
Preprocessing followed the `nnU-Net` framework described in Section 3.1, with several task-specific modifications. First, spatial resampling was restricted to the axial slice direction, while the original in-plane sampling and image dimensions were retained. The slice spacing was adapted to 1.0 mm. Resampling along this direction followed the `nnU-Net` anisotropy handling: for strongly anisotropic images with

$$\frac{\max\{s_x, s_y, s_z\}}{\min\{s_x, s_y, s_z\}} > 3,$$

nearest-neighbor interpolation was used along the axial slice direction. Otherwise, Lanczos-3 interpolation (see Section 2.4) was applied. Binary segmentation masks were resampled using nearest-neighbor interpolation.

Second, the CT image intensities $I(\omega)$, $\omega \in \Omega$, were clipped to the interval $[-1000, 2000]$ Hounsfield units, shifted by subtracting 500, and subsequently divided by 500. Accordingly, the CT image normalization was defined as

$$I_{\text{norm}}(\omega) = \frac{\text{Clip}(I(\omega), -1000, 2000) - 500}{500}, \qquad \omega \in \Omega.$$

This transformation mapped the clipped intensity range to $[-3, 3]$.

The same normalization and spatial preprocessing procedure was used for `OneClick` inference, for the generation of the refinement training samples, and during inference with the proposed models.

During postprocessing, connected-component analysis is applied and only the component closest to the patch center, corresponding to the user-selected point, is retained. To separate potentially merged lesions, an ellipsoid is subsequently fitted to the selected component and used in a watershed-based splitting procedure.

3.3 nnInteractive

`nnInteractive` is a general-purpose interactive segmentation framework based on `nnU-Net` for three-dimensional biomedical images, introduced by Isensee et al. [20]. It was designed as an open-set instance segmentation model for 3D biomedical images. The model can segment previously unseen structures based on user-provided spatial guidance and enables iterative refinement of segmentations. The framework supports positive and negative point prompts, scribbles, two-dimensional bounding boxes, and lasso interactions. These interactions can be placed in axial, sagittal, or coronal views and are used to generate complete three-dimensional segmentation masks.

Network architecture and input representation.
`nnInteractive` is built directly on the `nnU-Net` framework described in Section 3.1. Rather than introducing an independent segmentation framework, it extends the `nnU-Net` architecture and training framework with spatial interaction inputs and iterative mask refinement. Its segmentation network uses the three-dimensional Residual Encoder U-Net in the large `nnU-Net` configuration, referred to as ResEnc-L [20, 19]. The ResEnc-L architecture contains six resolution stages and uses residual blocks in the encoder, while the decoder follows the conventional `nnU-Net` design with skip connections.

`nnInteractive` follows an early-prompting strategy in which the spatial interaction representations are concatenated with the image and previous-mask channels before the first encoder stage. The interaction input is represented by six spatial channels: separate positive and negative channels are used for point prompts together with scribbles, while bounding boxes and lasso interactions share another pair of positive and negative channels each. The previous segmentation prediction is provided through an additional mask channel. Image information, the current segmentation state, and the user guidance are processed jointly throughout the complete encoder–decoder network rather than combining the prompts with an already computed latent image representation. Intensity normalization is applied only to the image channel, whereas the previous-mask and interaction channels are provided without intensity normalization.

Training data and target construction.
`nnInteractive` was trained on more than 120 publicly available volumetric biomedical segmentation datasets comprising 64 518 image volumes and 717 148 annotated object instances. Five percent of the image volumes were retained for internal validation. The

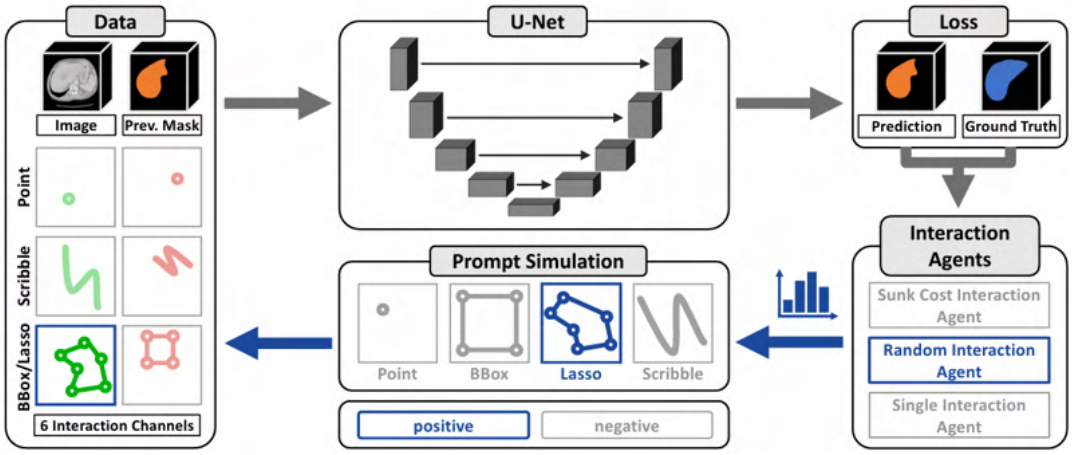


Figure 6: **Overview of the `nnInteractive` training pipeline.** Reproduced from Isensee et al. [20], Figure 2, licensed under CC BY 4.0. During training, user interactions are simulated dynamically based on the current model prediction and ground-truth segmentation. Thus, the error regions encountered during training and the resulting interaction sequences depend on the evolving model outputs rather than on a predefined set of correction samples.

training collection covers multiple imaging modalities, including CT, MRI, PET, three-dimensional ultrasound, and three-dimensional microscopy, as well as a wide range of anatomical and pathological structures.

The model was trained for instance segmentation rather than semantic segmentation. Semantic segmentations were therefore converted into individual object instances using connected-component analysis. To further increase the diversity of training targets, additional pseudo-instances, referred to as SuperVoxels, were generated using high-confidence SAM segmentations and SAM2-based propagation across image slices. During patch sampling, pseudo-instances were selected with a probability of 0.2.

Dynamic interaction simulation during training.
During training, `nnInteractive` simulates user interactions dynamically rather than relying on a predefined set of interaction samples. Starting from an initial interaction, the network generates a segmentation prediction, which is compared with the corresponding ground-truth segmentation. Based on the resulting segmentation errors, a subsequent simulated user interaction is generated according to the selected interaction agent. The updated interaction information is then provided to the network for another prediction. Consequently, the error regions encountered during training and the resulting interaction sequences depend on the current model prediction. The overall training procedure is provided in Figure 6.

To simulate different user behaviors during training, `nnInteractive` employs three interaction agents. Since this thesis considers only point-based interactions, the individual

agent strategies are not discussed further.

Error-component and point-prompt simulation.
For the spatial placement of a simulated interaction, the current prediction M is compared with the corresponding reference segmentation R. From the voxel-wise differences, include and exclude error regions are identified. Using the notation introduced in Section 2, these correspond to

$$\mathcal{E}^+ = \mathcal{R} \setminus \mathcal{M}, \qquad \mathcal{E}^- = \mathcal{M} \setminus \mathcal{R}.$$

The error regions are decomposed into connected error components. An error component $\mathcal{C}$ is then sampled with probability proportional to its number of voxels. Thus, larger components are more likely to be selected, while smaller components may still be chosen. A component from $\mathcal{E}^+$ induces a positive interaction, whereas a component from $\mathcal{E}^-$ induces a negative interaction.

For a point interaction, `nnInteractive` calculates a normalized Euclidean distance transform

$$D : \mathcal{C} \to [0, 1]$$

within the selected error component $\mathcal{C}$. Voxels farther from the component boundary receive larger distance-transform values. A prompt location is sampled according to

$$\mathbb{P}(p = x \mid \mathcal{C}) = \frac{D(x)^\alpha}{\sum_{z \in \mathcal{C}} D(z)^\alpha}, \qquad x \in \mathcal{C}.$$

The implementation uses either $\alpha = 8$ for strongly center-biased sampling or $\alpha = 1$ for the less center-biased sampling variant.

After sampling, the point is expanded into a three-dimensional spherical region. A second Euclidean distance transform is calculated inside this region and normalized by its maximum value. The resulting soft-valued prompt sphere has its maximum intensity at the sampled point, with decreasing values towards the boundary of the spherical support. Multiple point prompts assigned to the same interaction channel are combined using the voxel-wise maximum.

When a follow-up interaction is added, previously provided interactions are retained, but their intensities are multiplied by 0.9. The updated interaction maps are provided together with the latest model prediction for the subsequent refinement step. Thus, the interaction sequence is generated dynamically and depends on both the current prediction and the accumulated interaction history.

Preprocessing and inference.
For inference in this thesis, `nnInteractive` was accessed through a MEVIS-internal implementation in MeVisLab 4.2.70. The input data were resampled to the isotropic target spacing $\tilde{s} = (1.5, 1.5, 1.5)\,\text{mm}$. The implementation follows the anisotropy handling of `nnU-Net` [19]. For CT intensities, Lanczos-3 interpolation (see Section 2.4) was used instead of third-order spline interpolation used in standard `nnU-Net` preprocessing.

During interactive inference, point-prompt coordinates specified in physical image space were transformed to the voxel grid of the preprocessed image and rounded to the nearest voxel position. Previously provided interactions were retained in a prompt cache and were therefore available for subsequent refinement steps.

The spatial extent of the inference region was adapted to the accumulated prompt information. Let e_d denote the extent of the prompt representation along spatial dimension d. The corresponding inference region size was set to

$$n_d = \max\{192, e_d + 16\}.$$

Consequently, the inference region had a minimum spatial size of $192 \times 192 \times 192$ voxels and was enlarged when the accumulated prompts extended beyond this region. Inference regions exceeding the network processing size were processed tile-wise.

The current segmentation state was retained between interactions and provided as the previous-mask input for subsequent model updates. The MEVIS-internal implementation used in this thesis did not employ the AutoZoom mechanism described for the original `nnInteractive` framework [20].

Use in this thesis.
In this thesis, `nnInteractive` was used as the general-purpose interactive segmentation baseline for comparison with the proposed MultiClick models. Only positive (include) and negative (exclude) point prompts were considered to provide a consistent interaction setting across all evaluated models.

For error-guided refinement, the current segmentation was compared with the corresponding reference mask, connected error components were identified, and the next prompt was placed according to the experiment-specific component- and point-selection rules described in Section 6.2.

Chapter 4: Data

This work uses two closely related datasets for different purposes. Firstly, a restricted MEVIS-internal dataset is used to evaluate `nnInteractive` and the proposed refinement models. It contains initial masks generated by the `OneClick` Lesion Segmentation tool and corresponding masks manually corrected by a radiologist. This combination enables the evaluation of refinement models on segmentation errors that were actually corrected during a radiological annotation workflow. Secondly, refinement training data was generated from the publicly available Longitudinal-CT dataset from the University Hospital Tübingen [24, 12]. Both datasets consist of whole-body CT scans of patients with melanoma.

A single lesion mask may require corrections at multiple spatially separated locations. To identify these distinct error regions, connected-component analysis was applied to the voxel-wise differences between the initial and reference mask. All mask differences, connected components, and component sizes were calculated on the original voxel grids without additional resampling.

Three-dimensional 26-connectivity was used to define connected components, meaning that voxels sharing a face, edge, or corner were considered connected. This choice reduces artificial fragmentation by preventing diagonally adjacent error voxels from being interpreted as separate single-voxel or very small components. A more detailed description is provided in Section 2.3.

4.1 Evaluation Dataset

The restricted MEVIS-internal dataset was created during a validation process of the `OneClick` lesion segmentation tool, which is further described in Section 3.2. The purpose of this validation was to assess the practical usefulness of the tool in a clinical annotation setting. A radiologist operated the `OneClick` lesion segmentation tool by selecting a lesion and placing a point prompt to generate its initial segmentation mask. If a generated mask was not sufficiently accurate, it was manually corrected. These corrections were performed using a two-dimensional circular structuring element with an adjustable radius and automatic interpolation between manually annotated axial slices. The resulting dataset enables the analysis of the frequency and characteristics of manual segmentation corrections.

The dataset is also well suited for evaluating interactive refinement methods. Since it contains both the initial model-generated mask and the corresponding manual corrections by a radiologist, it provides realistic examples of segmentation errors that occur during a clinical annotation workflow. This allows refinement models to be evaluated on correction patterns observed in the clinical annotation workflow. The manually corrected masks obtained after the review process are used as reference segmentations in this thesis. An unchanged lesion denotes a `OneClick` segmentation for which no manual correction was recorded during the annotation workflow.

The evaluation dataset was also used for exploratory error analysis. Insights into

the composition of the error components informed the qualitative design of the filtering strategy, whereas the numerical filtering thresholds were not optimized on the evaluation dataset. In this context, filtering refers to the removal of error components with predefined characteristics from the training data. For example, error-component characteristics that occurred frequently in the evaluation dataset were not used as exclusion criteria. Therefore, results obtained on this dataset should be interpreted as exploratory rather than as a performance assessment in a fully independent test setting.

Dataset composition.
Table 1 provides an overview of the evaluation dataset composition. Overall, the dataset contains 93 processed CT image volumes from 41 patients with melanoma, comprising 47 baseline and 46 follow-up volumes. Some original CT examinations were divided into multiple image volumes because of their large number of axial slices.

At least one lesion was corrected in 67 of the 93 processed CT image volumes. Thus, although no manual correction was recorded for most individual lesion masks (Table 2), manual refinement was performed for at least one lesion in the majority of processed image volumes.

Characteristic	**Value**
Patients	41
Processed CT image volumes	93
Image volumes with at least one corrected lesion	67 (72.0%)

Table 1: **Composition of the restricted MEVIS-internal evaluation dataset at patient and image-volume levels.**

Lesion characteristics and correction status.
Table 2 summarizes the evaluation dataset at the lesion-mask level. Overall, the dataset contains 2494 lesion masks, of which 248 were manually corrected, corresponding to approximately 9.9%. For the remaining 2246 masks, no correction was recorded.

Include regions are regions present in the reference mask but absent from the automatic initial mask, whereas *exclude regions* are present in the automatic initial mask but absent from the reference mask. Among the corrected lesion masks, 83 contained include-only corrections, 89 contained exclude-only corrections, and 76 contained both include and exclude corrections.

The mask-size and volume statistics reported in Table 2 refer exclusively to these 248 corrected lesion masks. The initial masks had a mean volume of $7.169\,\mathrm{cm}^3$ and a median volume of $0.986\,\mathrm{cm}^3$. The corresponding reference masks had a mean volume of $13.902\,\mathrm{cm}^3$ and a median volume of $1.054\,\mathrm{cm}^3$. For both mask types, the mean was substantially larger than the median, indicating strongly right-skewed lesion-size distributions influenced by a comparatively small number of larger lesions. Representative lesion masks covering a range from very small to large lesions are shown in Figure 7.

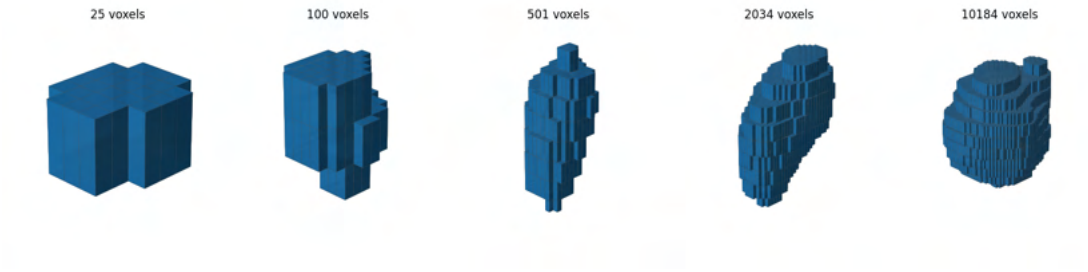


Figure 7: **Representative lesion masks of increasing size from the evaluation dataset.** The examples range from 25 to 10 184 voxels and illustrate the substantial variation in lesion size represented in the dataset.

Characteristic	**Value**
Total lesion masks	2494
Lesion masks without recorded correction	2246
Corrected lesion masks	248
Corrected lesion masks [%]	9.9
Include-only lesion masks	83
Exclude-only lesion masks	89
Include-and-exclude lesion masks	76
Corrected lesion masks only	
Mean initial mask size [voxels]	3288.8
Median initial mask size [voxels]	453.5
Mean reference mask size [voxels]	6041.1
Median reference mask size [voxels]	458.0
Mean initial mask volume [cm^3]	7.169
Median initial mask volume [cm^3]	0.986
Mean reference mask volume [cm^3]	13.902
Median reference mask volume [cm^3]	1.054

Table 2: **Lesion-mask statistics of the restricted MEVIS-internal evaluation dataset.** Overall lesion counts are reported for the complete dataset, whereas mask-size and volume statistics are reported for the 248 corrected lesion masks only. Initial masks refer to the `OneClick` segmentations, whereas reference masks refer to the masks obtained after radiological review. Physical mask volumes were calculated using the voxel geometry of the corresponding NIfTI image.

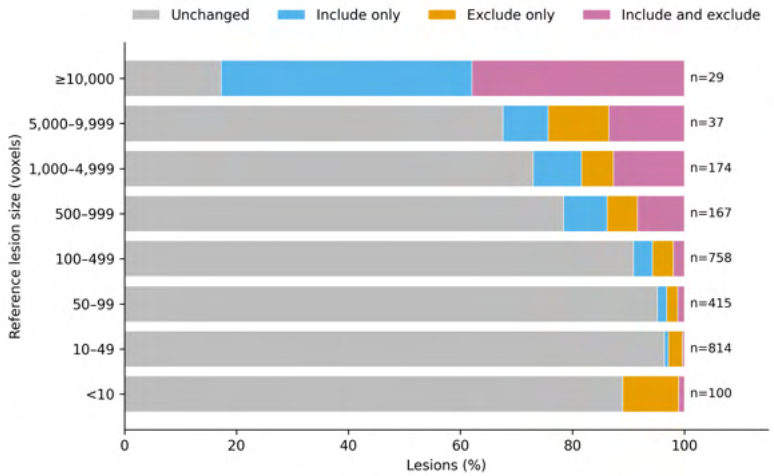


Figure 8: **Distribution of correction status across reference lesion-size classes in the restricted MEVIS-internal evaluation dataset.** Bars show the relative proportions of unchanged, include-only, exclude-only, and include-and-exclude lesions within each size class. The number of lesions in each class is shown on the right. The proportion of corrected lesions generally increases with lesion size, with particularly high correction rates among the largest lesions.

Figure 8 illustrates the distribution of correction types across different reference lesion-size classes. Only a small proportion of lesions containing 10–499 voxels were manually corrected. The proportion of corrected lesions increased in the larger lesion-size classes, except for the smallest category. In particular, corrections were recorded for more than 80% of lesions containing at least 10 000 voxels. One possible explanation is the fixed patch size of the `OneClick` tool, which can limit the segmentation of large lesions. The smallest lesion-size category showed a comparatively high proportion of exclude-only corrections.

The dataset covers different types of metastatic lesions, including lung, lymph-node, and liver lesions. As illustrated in Figure 9, lung lesions were the most frequent category with 1135 lesion samples, followed by liver lesions with 502 cases and lymph-node lesions with 296 cases. Relative to their frequency, lung and liver lesions were corrected infrequently, whereas corrections were recorded for more than 20% of lymph-node lesion masks. Skeletal lesions had the highest correction rate. For all lesion types represented by more than 100 cases, both include and exclude corrections occurred.

Error component statistics.
Component sizes were quantified both as voxel counts and as physical volumes based on the voxel geometry of the corresponding NIfTI image. The 248 corrected lesion masks resulted in 590 connected error components, including 317 include and 273 exclude components. The component-size distributions were strongly right-skewed. Include components had a mean volume of 6.025 cm^3 and a median volume of 0.054 cm^3, whereas

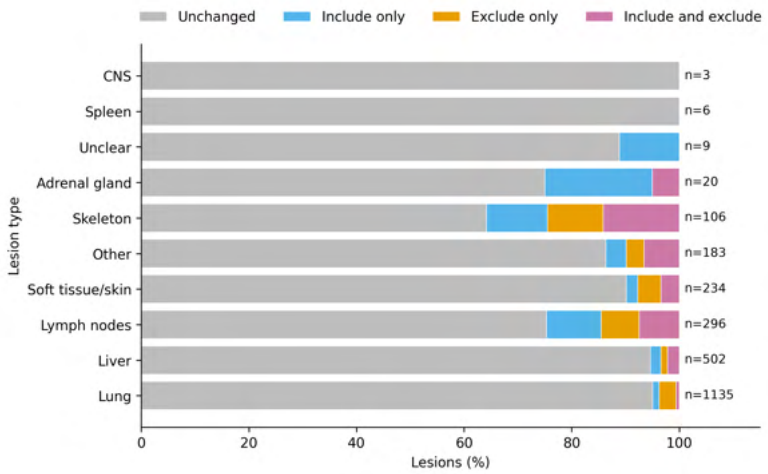


Figure 9: **Distribution of lesions by lesion type and correction status in the restricted MEVIS-internal evaluation dataset.** For unchanged lesion masks, no manual correction was recorded. Include-only lesions contained connected components that were added to the segmentation mask by the radiologist, whereas exclude-only lesions contained connected components that were removed from the mask. Lesions with both include and exclude changes contained both added and removed connected components. Lung and liver lesions were predominantly unchanged, whereas lymph-node and skeletal lesions showed comparatively higher proportions of corrected masks.

Characteristic	**Value**
Total error components	590
Include components	317
Exclude components	273
Mean include component size [voxels]	2484.2
Median include component size [voxels]	27.0
Mean include component volume [cm^3]	6.025
Median include component volume [cm^3]	0.054
Mean exclude component size [voxels]	384.4
Median exclude component size [voxels]	34.0
Mean exclude component volume [cm^3]	0.879
Median exclude component volume [cm^3]	0.071
Components smaller than 10 voxels	204
Components larger than 500 voxels	83
Single-axial-slice components	363 (61.5%)
Include components	190
Exclude components	173

Table 3: **Statistics of connected error components in the restricted MEVIS-internal evaluation dataset.** Component sizes are reported as voxel counts and as physical volumes calculated using the voxel geometry of the corresponding NIfTI image.

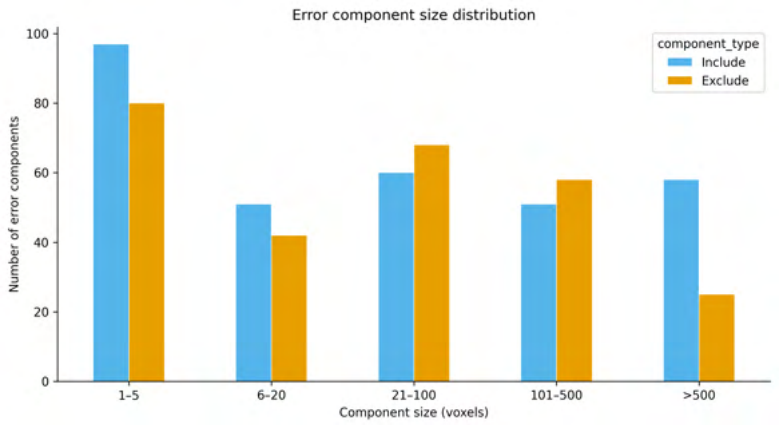


Figure 10: **Distribution of connected error-component sizes in the restricted MEVIS-internal evaluation dataset.** Blue bars represent include components that were added during manual correction, whereas orange bars represent exclude components that were removed from the initial segmentation. Most components contained fewer than 100 voxels, although larger components occurred for both correction types.

exclude components had a mean volume of 0.879 cm^3 and a median volume of 0.071 cm^3.

Overall, 363 error components (61.5%) were restricted to a single axial image slice. These comprised 190 include components and 173 exclude components. In addition, 204 components contained fewer than 10 voxels, whereas 83 components contained more than 500 voxels. Figure 10 presents the error-component distribution across component-size classes in greater detail. Most error components contained fewer than 100 voxels.

To account for differences in lesion size, the relative component size was calculated by dividing the voxel count of each error component by the size of the corresponding reference lesion mask. The same normalization was applied to include and exclude components, enabling a direct comparison between both error types. Figure 11 shows the resulting distribution. Although many error components were small in absolute terms, some represented a substantial proportion of the corresponding reference lesion.

Shapes of error components.

The geometry of connected error components was characterized using principal component analysis (PCA). PCA identifies orthogonal principal directions from eigenvectors of a covariance matrix, while the corresponding eigenvalues quantify the variance along these directions [21]. The descriptors were based on the principal-moment ratios used for label-shape analysis in ITK [18]. In this thesis, the ratios were expressed in reciprocal form such that their values are bounded between zero and one. Let $\mathcal{C} = \{\omega_1, \ldots, \omega_N\} \subseteq \Omega$ be a connected error component. Using the physical voxel position $x_j = x(\omega_j)$ as defined in Section 2.1 (Equation 2), the physical centroid of the component is given by

$$\bar{x} = \frac{1}{N} \sum_{j=1}^{N} x_j.$$

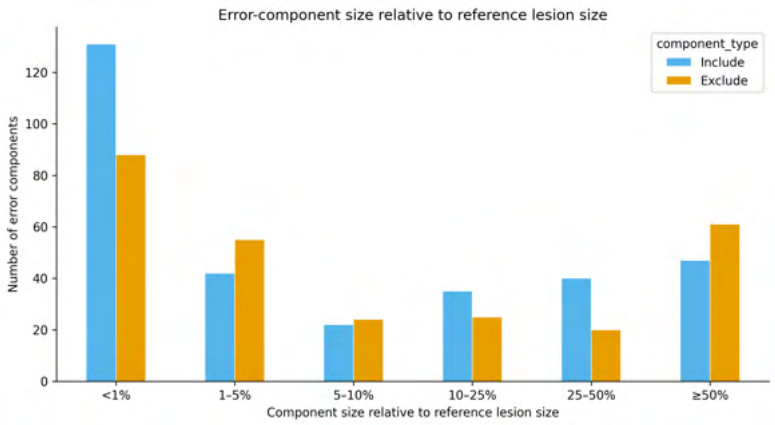


Figure 11: **Distribution of connected error-component sizes relative to the reference mask in the restricted MEVIS-internal evaluation dataset.** Blue bars represent include components that were added during manual correction, whereas orange bars represent exclude components that were removed from the initial segmentation. Most error components account only for a small fraction of the corresponding reference lesion, while large relative errors occur less frequently.

The covariance matrix of the physical voxel-center distribution is defined as

$$\Sigma_{\mathcal{C}} = \frac{1}{N} \sum_{j=1}^{N} (x_j - \bar{x})(x_j - \bar{x})^{\mathsf{T}}.$$

Let

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$$

denote the eigenvalues of $\Sigma_{\mathcal{C}}$ in descending order. The square roots of the eigenvalues describe the spatial spread of the component along its three principal axes.

Linear elongation was defined as

$$E(\mathcal{C}) = \sqrt{\frac{\lambda_2}{\lambda_1}},$$

and **flatness** was defined as

$$F(\mathcal{C}) = \sqrt{\frac{\lambda_3}{\lambda_2}}.$$

Both descriptors are dimensionless and take values between zero and one. An elongation ratio close to one indicates similar spatial spreads along the first and second principal axes, whereas a small value indicates a dominant first principal axis and therefore a more elongated component. A flatness ratio close to one indicates similar spatial spreads along the second and third principal axes, whereas a small value indicates a smaller spread along the third principal axis and thus a more planar component. Representative error components with different PCA-based shape characteristics are shown in Figure 12.

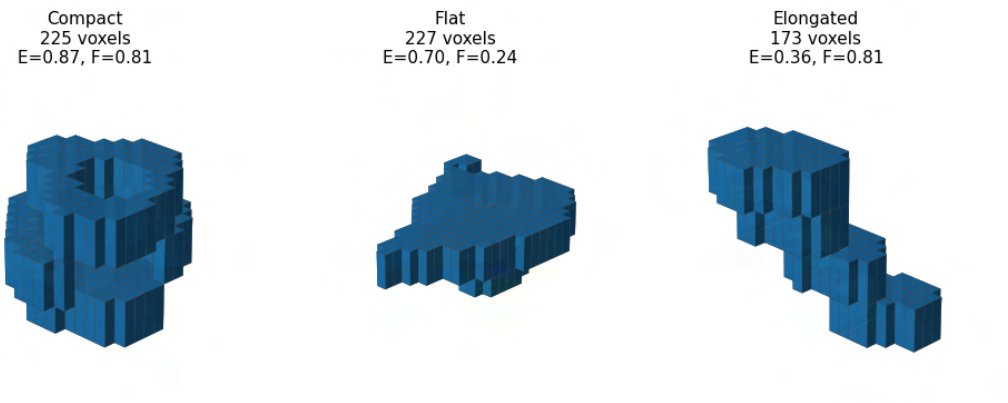
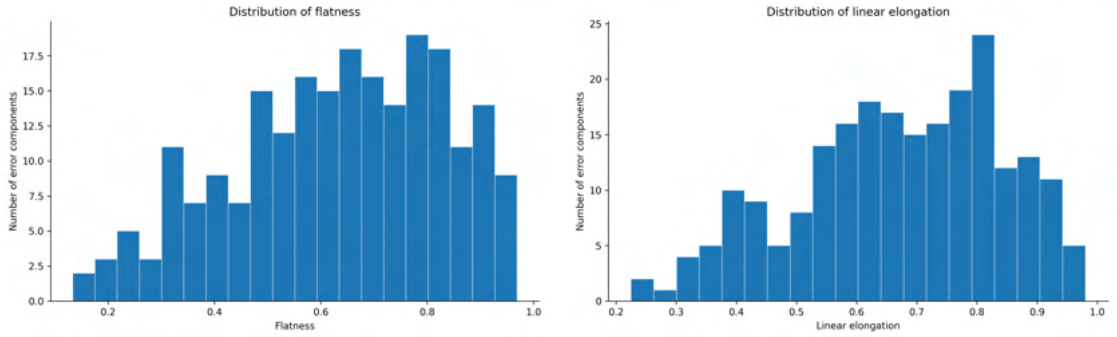


Figure 12: **Representative error components with different PCA-based shape characteristics from the evaluation dataset.** The compact component exhibits comparable spreads along the three principal axes. The flat component has a comparatively small third principal spread, resulting in a low flatness ratio F. The elongated component is dominated by one principal direction, resulting in a low elongation ratio E. The displayed ratios were calculated using physical voxel coordinates.

PCA-based shape metrics were available for 386 components comprising at least 10 voxels. Components below this size were excluded to reduce sensitivity to voxel discretization and unstable PCA estimates for very small components. For the visualization in Figure 13, components restricted to a single axial slice were excluded, since their zero extent in the third spatial direction results in a flatness value of zero and would dominate the flatness distribution. The distributions indicate a broad range of three-dimensional error-component geometries rather than being concentrated around a single characteristic shape.

Association between component size and shape descriptors.
Since PCA-based shape descriptors may depend on the spatial extent of an error component, their association with component size was examined using Spearman rank correlation [1]. Spearman's rank correlation was used due to the strongly right-skewed distribution of component size and its ability to capture non-linear monotonic associations. In the following, $\rho \in [-1, 1]$ denotes the Spearman's rank correlation coefficient, with its sign indicating the direction and its magnitude the strength of the monotonic association. The corresponding p-value quantifies the evidence against the null hypothesis of no monotonic association.

Considering all components with valid shape descriptors, component size showed a moderate positive correlation with the linear-elongation ratio ($\rho = 0.30$, $p < 0.001$) and a stronger correlation with the flatness ratio ($\rho = 0.61$, $p < 0.001$). However, after



(a) Distribution of the flatness ratio.

(b) Distribution of the linear elongation ratio.

Figure 13: **Distributions of PCA-based shape descriptors for multi-slice error components.** Shape descriptors were calculated only for components containing at least 10 voxels to reduce sensitivity to voxel discretization and unstable PCA estimates for very small components. Components restricted to a single axial slice were excluded from the distributions. Smaller values correspond to stronger flatness in (a) and stronger elongation in (b). Both distributions cover a broad range of values, while strongly flat or elongated components with low metric values occur less frequently.

excluding components restricted to a single axial slice, both associations were weak, with $\rho = 0.11$ ($p = 0.095$) for the linear-elongation ratio and $\rho = 0.07$ ($p = 0.293$) for flatness. Thus, both corresponding p-values exceeded 0.05. This indicates that the apparent association between component size and PCA-based shape descriptors, particularly flatness, was largely driven by single-slice error components.

Limitations of the dataset.

The evaluation dataset has four main limitations. First, it contains only patients with melanoma and is dominated by lung lesions, whereas other lesion types are less represented. This may limit generalizability of the evaluation results to other tumor entities and clinical workflows.

Second, the initial point prompt was placed by a single radiologist, who also reviewed and manually corrected the resulting `OneClick` segmentation. Therefore, the reference masks were not independently reviewed by a second radiologist.

Third, the exact locations of the point prompts used for the `OneClick` tool and the sequence of subsequent mask corrections were not documented. As a result, individual user interactions cannot be reconstructed in detail, and the potential influence of prompt placement and correction order cannot be evaluated. Consequently, the experiments do not attempt to reproduce the original user interactions. Instead, standardized correction prompts are simulated from the observed error components as described in Section 6.2.

Fourth, the distribution of observed error components is conditional on the performance of the `OneClick` model used to generate the initial masks. The observed corrections reflect not only lesion characteristics but also model-specific failure modes. As a result, error components may be overrepresented for lesion types on which the model

performs poorly and underrepresented for lesion types on which it performs well. Therefore, the observed error distribution should not be interpreted as an unbiased estimate of the general prevalence of segmentation errors across lesion types.

The evaluation results should therefore be interpreted in the context of the specific dataset composition, model, and annotation workflow.

Implications for model evaluation.
Although manual correction was not recorded for most individual lesion masks, at least one lesion mask was corrected in 67 of the 93 processed image volumes. The dataset represents an annotation workflow in which many initial segmentations remained unchanged, while targeted refinement was still performed on the majority of processed image volumes.

The recorded corrections consist of both include and exclude connected error components and vary substantially in size and geometry. In particular, a refinement method must handle corrections ranging from a few voxels to large error regions, error components with varying degrees of flatness and elongation, as well as components restricted to a single axial slice. Very small lesions require particular care because changes in only a few voxels can represent a large relative change in mask volume.

Overall, these characteristics make the dataset suitable for evaluating whether an interactive refinement method can correct diverse localized errors while preserving already correct parts of the initial segmentation.

4.2 Training Dataset

The publicly available Longitudinal-CT dataset [24, 12] contains 600 whole-body CT examinations from 300 patients with melanoma. For each patient, the dataset provides two examinations acquired at different time points: a baseline examination and a follow-up examination during systemic therapy at University Hospital Tübingen.

Annotation procedure.
The lesion annotations were created using a two-stage annotation procedure. First, one radiologist with extensive experience in oncologic imaging initially segmented all visible primary tumors and metastases on axial CT slices manually. The segmentations were created from scratch using the web-based CuraMate annotation platform. An adjustable brush tool was used to define lesion contours and geometric interpolation could be applied to generate contours on intermediate slices. Baseline and follow-up examinations were displayed side by side to identify corresponding lesions. In a second stage, the initial annotations were reviewed by two experienced radiologists, who reached a consensus on the final lesion masks.

Derived refinement dataset.
The original Longitudinal-CT dataset does not contain initial segmentation masks or recorded correction steps. Consequently, error components are not directly available in the source dataset. For the training data used in this thesis, initial lesion masks were generated using the `OneClick` tool and compared with the corresponding reference

masks, which served as target segmentations. The resulting voxel-wise differences were decomposed into connected include and exclude error components. The complete generation procedure is described in Section 5.1. This section summarizes the resulting data composition.

Lesion characteristics.
The values in Table 4 indicate that almost all generated initial masks differed from their reference masks. The dataset contained 7170 reference lesion masks. For 7156 masks (99.8%), the generated initial mask differed from the corresponding reference mask. Among these masks, 468 contained include regions only, 581 contained exclude regions only, and 6107 contained both error types. Only 14 generated masks were identical to their corresponding reference masks.

The generated initial masks achieved a mean Dice score of 0.789 and a median Dice score of 0.835. The comparatively high initial overlap may partly be explained by the fact that the `OneClick` model was trained using baseline data from the Longitudinal-CT dataset. These values characterize the generated training data and should not be interpreted as an independent performance assessment of the `OneClick` model.

Characteristic	**Value**
Total reference lesion masks	7170
Initial masks containing generated errors	7156
Initial masks identical to the reference mask	14 (0.2%)
Include-only lesion masks	468
Exclude-only lesion masks	581
Include and exclude lesion masks	6107
Mean Dice score	0.789
Median Dice score	0.835
Masks containing generated errors	
Mean initial mask volume [cm^3]	3.600
Median initial mask volume [cm^3]	0.350
Mean reference mask volume [cm^3]	8.016
Median reference mask volume [cm^3]	0.361

Table 4: **Characteristics of lesions in the derived refinement training dataset.** Mask-volume statistics were calculated for the 7156 lesion masks containing generated errors. Initial masks refer to the `OneClick` segmentations, whereas reference masks refer to the manual annotations provided by the Longitudinal-CT dataset. Almost all `OneClick` initializations differed from their reference masks, and most erroneous masks contained both include and exclude errors.

Figure 14 shows the distribution of lesion types in the processed dataset. Similar to the evaluation dataset composition of lesion types shown in Figure 9, the same four

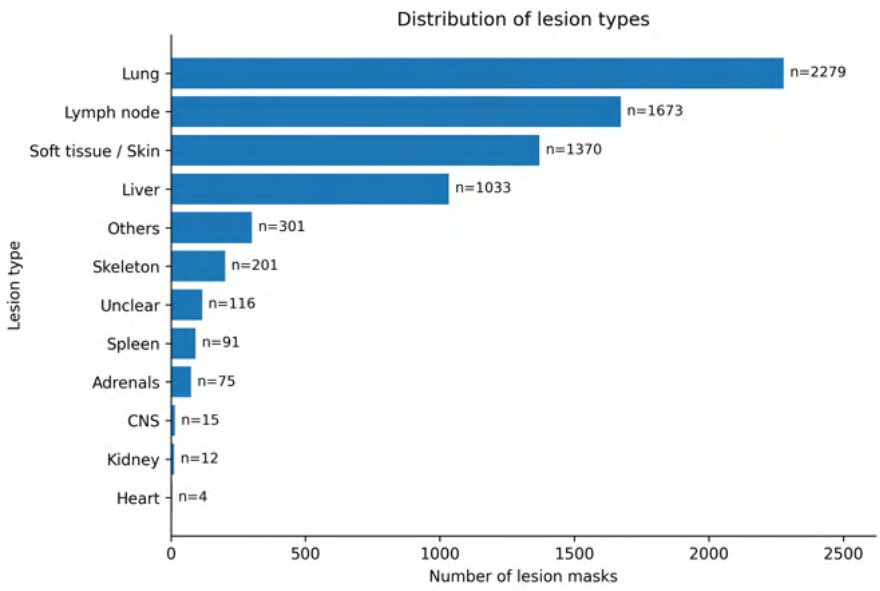


Figure 14: **Absolute frequencies for lesion types in the Longitudinal-CT dataset.** Lung lesions form the largest category, followed by lymph-node, soft-tissue or skin, and liver lesions.

lesion categories were most frequent. Lung lesions formed the largest category, with 2279 lesion masks, followed by lymph-node lesions with 1673 masks, soft-tissue or skin lesions with 1370 masks, and liver lesions with 1033 masks. Several other lesion types were represented by substantially fewer cases. Consequently, the training dataset is dominated by the same four lesion categories as the evaluation dataset.

As illustrated in Figure 15, most lesion masks contained fewer than 500 voxels, with the largest individual size class comprising lesions between 100 and 499 voxels. At the same time, 327 reference masks contained at least 10 000 voxels.

Error component statistics.

Table 5 summarizes the error-component characteristics of the derived training data. To facilitate direct comparison with the evaluation dataset, include components correspond to missing regions in the initial mask, whereas exclude components correspond to false-positive regions. The voxel-wise comparison of the generated initial masks and the corresponding reference masks resulted in 72 197 connected error components. Include and exclude components occurred with similar frequencies, accounting for 36 863 and 35 334 components, respectively.

Include components had a mean volume of 0.937 cm^3 but a median volume of 0.0046 cm^3. Exclude components had a mean volume of 0.083 cm^3 and a median volume of 0.0047 cm^3. The similar medians indicate that the typical component was very small for both error types, whereas the larger mean volume of include components was influenced by a small number of large include regions. This difference may partly result from the fixed patch

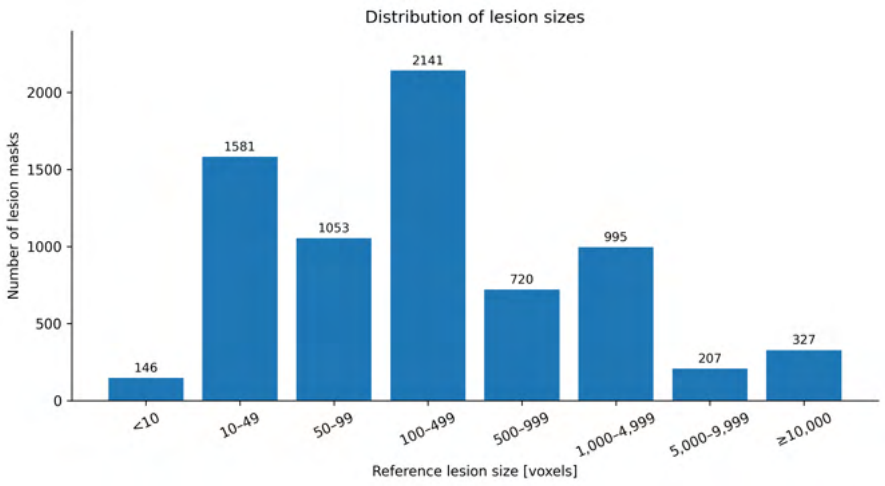


Figure 15: **Absolute frequencies for lesion size classes of the training dataset.** The lesion-size distribution covered several orders of magnitude and was dominated by comparatively small lesions with a size smaller than 500 voxels.

size used by the `OneClick` model.

A graphical overview of the absolute component-size distribution is provided in Figure 16a. Overall, 56 151 components (77.8%) contained fewer than 10 voxels, and 52 342 components (72.5%) were restricted to a single axial image slice. The generated training dataset is therefore dominated by small and spatially localized error components. Nevertheless, 1182 components contained more than 500 voxels, ensuring that the dataset also includes larger correction regions.

Relative component sizes were calculated using the same definition as for the evaluation dataset: the size of each error component was divided by the size of the corresponding reference lesion mask. The resulting distribution is shown in Figure 16b. The median relative component size was approximately 0.32% for both error types. Thus, the typical connected component represented only a small fraction of the corresponding lesion mask, although a smaller number of components accounted for a substantial part of the mask. This distinction is relevant for training-data filtering because a component containing only a few voxels can still represent an important correction for a very small lesion.

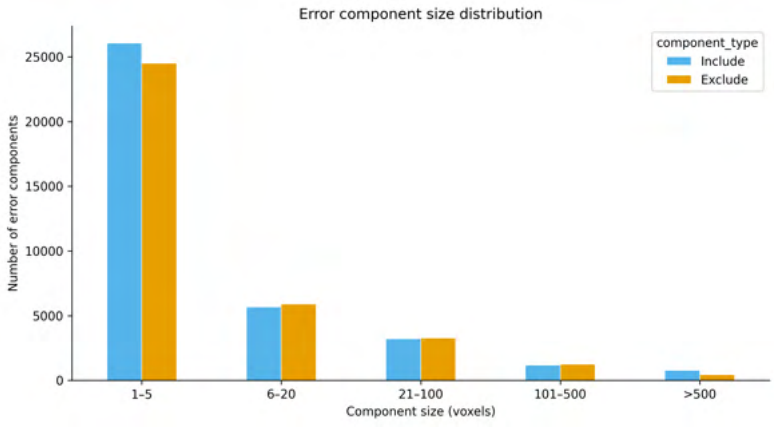
PCA-based shape descriptors were calculated for 16 046 components containing at least 10 voxels. After excluding components restricted to a single slice, 8 858 multi-slice components remained for the distributions shown in Figure 17. The remaining error components show a broad range of three-dimensional shapes.

Limitations of the training data.

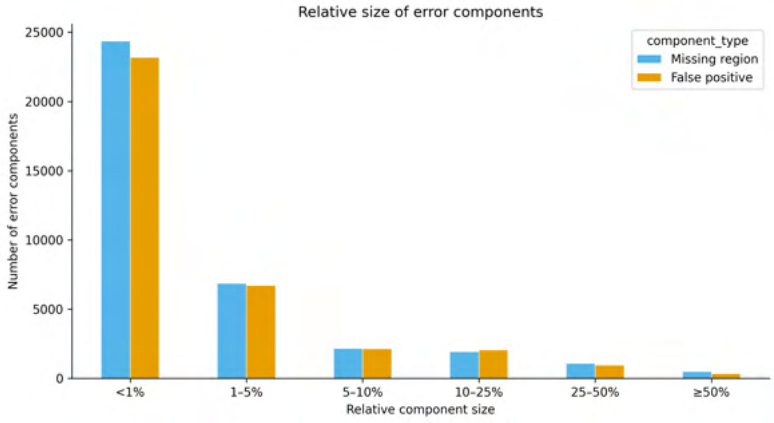
The Longitudinal-CT dataset does not contain imperfect initial segmentation masks or recorded human correction sequences. Therefore, the initial masks and error components used for training were generated synthetically using the `OneClick` model. Their distribution depends on the selected generation procedure and does not necessarily reproduce

Characteristic	Value	
Component composition		
Total error components	72 197	
Include components (missing regions)	36 863	(51.1%)
Exclude components (false-positive regions)	35 334	(48.9%)
Component volume		
Mean include component volume [cm^3]	0.937	
Median include component volume [cm^3]	0.0046	
Mean exclude component volume [cm^3]	0.083	
Median exclude component volume [cm^3]	0.0047	
Size and slice extent		
Components smaller than 10 voxels	56 151	(77.8%)
Components larger than 500 voxels	1182	(1.6%)
Single-axial-slice components	52 342	(72.5%)
Shape characteristics		
Components with at least 10 voxels	16 046	(22.2%)
Multi-slice components with at least 10 voxels	8858	(12.3%)

Table 5: **Characteristics of the connected error components generated for refinement-model training.** Components were extracted from the voxel-wise differences between the generated initial masks and the reference masks. Percentages refer to all 72 197 error components.

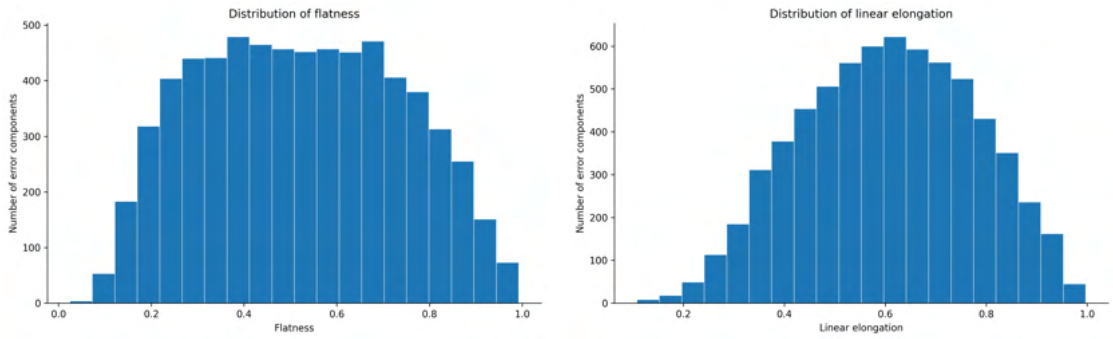


(a) Absolute error-component sizes.



(b) Error-component sizes relative to the reference lesion.

Figure 16: **Distributions of absolute and relative connected error-component sizes in the derived refinement training data.** (a) Absolute component sizes in voxels. (b) Component sizes relative to the corresponding reference lesion mask. Both distributions are dominated by small error components, while a smaller number of components are large either in absolute terms or relative to the corresponding lesion.



(a) Distribution of the flatness ratio.

(b) Distribution of the linear elongation ratio.

Figure 17: **Distributions of PCA-based shape descriptors for multi-slice error components.** Shape descriptors were calculated only for components containing at least 10 voxels to reduce sensitivity to voxel discretization and unstable PCA estimates for very small components. Components restricted to a single axial slice were excluded from the distributions. Smaller values correspond to stronger flatness in (a) and stronger elongation in (b). Both distributions cover a broad range of values, while strongly flat or elongated components with low metric values occur less frequently.

the distribution of interaction sequences in a real clinical annotation workflow.

Furthermore, the source dataset contains only patients with melanoma and is dominated by lung, lymph-node, soft-tissue, and liver lesions. The resulting training data may represent rare lesion types less comprehensively.

4.3 Summary

The analyses showed that both datasets contain lesions with a wide range of sizes and error components with different spatial characteristics. The error-component distributions were dominated by small and localized components, many of which were restricted to a single axial slice. Nevertheless, both datasets also contained larger error regions with diverse geometries, ranging from strongly planar or elongated shapes to more compact components. Furthermore, filtering decisions for the training data should consider the component size relative to the corresponding lesion mask rather than relying only on an absolute voxel threshold.

The two datasets also have important limitations. Both contain only patients with melanoma and are dominated by a small number of common lesion types, which may limit generalizability to other tumor entities. Furthermore, the correction patterns depend on the performance of the `OneClick` model. In the evaluation dataset, the observed corrections reflect the model-specific errors that were identified and corrected by a radiologist. In the training dataset, the error components depend on the procedure used to generate the initial masks. Finally, because the evaluation dataset informed the design of the training-data filtering, it does not represent a fully independent test setting.

Despite these limitations, the datasets provide a suitable basis for the aims of this

thesis. The evaluation dataset contributes clinically motivated correction examples, whereas the Longitudinal-CT dataset provides the scale required for model training. Together, they support the development and exploratory evaluation of interactive lesion-refinement models that must handle diverse error types while preserving already correct regions of an initial lesion segmentation.

Chapter 5: Proposed MultiClick Models for Interactive Lesion Segmentation

This chapter introduces four task-specific models for point-based interactive lesion segmentation and refinement in CT images, referred to as `MultiClick` models. They were developed to investigate whether training specifically on lesion-segmentation and correction examples can provide a more suitable refinement behavior than a general-purpose interactive segmentation approach like `nnInteractive`. The models are intended to respond to correction prompts while preserving regions of an existing lesion mask that are already correct.

The `MultiClick` models retain several principles of `nnInteractive`. Both approaches build on the `nnU-Net` framework, incorporate spatial interaction maps through early prompting, receive a current segmentation mask as input, and retain previous interactions through decaying prompt representations. However, the `MultiClick` models were designed specifically for lesion segmentation and refinement in CT images.

A central difference concerns the generation of training interactions. While follow-up interactions were dynamically derived from the current model prediction during the training process of `nnInteractive`, the `MultiClick` approach uses predefined interaction sequences generated before model training. Initial `OneClick` segmentations, connected error components, simulated point prompts, and intermediate mask states were therefore determined independently of the evolving `MultiClick` predictions. This makes it possible to control the composition of refinement training data and the corresponding segmentation targets. The controlled design allows the model to be trained on desired refinement behavior: responding to correction prompts while minimizing unnecessary changes to already correct regions of the current segmentation. It also allows individual training-data design choices to be varied systematically while keeping the remaining model configuration unchanged.

Based on this, two training-data design choices are investigated: the definition of the training target and the filtering of refinement samples with certain characteristics. Their combination resulted in four `MultiClick` variants introduced in Section 5.2.

The main similarities and differences between `nnInteractive` and the `MultiClick` models are summarized in Table 6. The individual design choices are described in detail in the following sections.

5.1 Training-Data Generation

Recorded user-interaction sequences were not available for training. We therefore constructed controlled lesion-refinement sequences from imperfect segmentations and corresponding reference masks. To obtain reproducible and standardized simulated interactions, both error-component selection and point placement were deterministic. The training-data generation pipeline, including `OneClick` inference, simulated interaction generation, and training-patch extraction, was implemented and executed in MeVisLab

Property	nnInteractive	MultiClick models
Intended scope	Biomedical general-purpose open-set 3D instance segmentation	Task-specific lesion segmentation and refinement in CT
Framework	nnU-Net with ResEnc-L backbone	nnU-Net with PlainConvUNet backbone
Interaction types	Positive and negative points, scribbles, boxes, and lassos	Positive and negative point prompts
Input strategy	Early prompting (prompt input channel)	Early prompting (prompt input channel)
Training interactions	Generated dynamically from the current model prediction	Generated before training using controlled mask updates
Training error-component selection	Random sampling proportional to component size	Largest error component
Training point selection	Probabilistic EDT-based sampling	Maximizer of EDT
Previous prompts	Retained and multiplied by 0.9	Retained and multiplied by 0.9
Training target	Complete selected object instance	Stepwise or final target
Preprocessing	Native voxel spacing and image-wise z-score normalization	Target spacing of $0.8 \times 0.8 \times 1.0$ mm and CT-specific normalization
Patch size	$192 \times 192 \times 192$ voxels	$128 \times 128 \times 128$ voxels
Training data	120+ heterogeneous 3D biomedical datasets comprising 64 518 images and 717 148 object instances	Longitudinal-CT dataset comprising 600 whole-body CT examinations from 300 melanoma patients

Table 6: **Comparison of the main design characteristics of `nnInteractive` and the proposed task-specific `MultiClick` models.** Both approaches build on `nnU-Net` and use spatial prompt maps together with a previous-mask input, but differ in their network configuration, preprocessing, interaction simulation, and training-target definition.

4.2.70. To describe the data generation procedure, the conventions introduced in Section 2 were used.

Generation of initial OneClick segmentations.
The publicly available Longitudinal-CT dataset, described in Section 4.2, served as the basis for generating the training data. Since the dataset contains reference lesion masks but no imperfect initial segmentations, an initial segmentation mask M_0 was generated for each reference lesion using the `OneClick` Lesion Segmentation model described in Section 3.2.

Let R denote the reference lesion mask and let

$$\mathcal{R} = \mathrm{fg}(R)$$

denote its foreground voxel set. The point prompt used to generate the initial `OneClick` segmentation was selected as a maximizer of the three-dimensional Euclidean distance transform (EDT) within $\mathcal{R}$:

$$p_{\mathrm{OC}} \in \arg\max_{p \in \mathcal{R}} d_{\mathrm{grid}}(p, \Omega \setminus \mathcal{R}).$$

The maximally interior voxel was chosen as a standardized simulated `OneClick` prompt that lies well within the reference lesion and is less sensitive to boundary uncertainty than a point close to the lesion surface.

The `OneClick` Lesion Segmentation model was applied to the corresponding CT image I using p_{OC}, yielding the initial segmentation mask M_0. Its foreground voxel set is denoted by

$$\mathcal{M}_0 = \mathrm{fg}(M_0).$$

Throughout this chapter, calligraphic symbols denote foreground voxel sets, whereas the corresponding non-calligraphic symbols denote binary mask images. In particular, for each interaction state t,

$$M_t = \mathbb{1}_{\mathcal{M}_t},$$

where $\mathbb{1}_{\mathcal{M}_t}$ denotes the binary indicator mask of $\mathcal{M}_t$. The resulting mask M_0 served as the initial input mask for the subsequent refinement-interaction sequence. The point p_{OC} was used only to generate M_0 and was not reused as an input prompt during the subsequent training-sample generation.

Based on the initial `OneClick` segmentation M_0, two types of training samples were generated: refinement samples representing individual correction steps and initial-segmentation samples representing segmentation from an empty input mask.

Image preprocessing and spatial alignment.
The `OneClick` inference internally applied the preprocessing described in Section 3.2. Its resulting mask was returned in the geometry of the corresponding input CT volume. Before subsequent preprocessing, the connected error components between the `OneClick` mask and the corresponding reference mask were analyzed on the original image grid to obtain the component-size information used for later sample filtering.

The CT image, reference mask, and initial `OneClick` mask were then preprocessed for the subsequent training-sample generation. The filtering criteria were evaluated on the connected error components on the original voxel grid, before spatial preprocessing. In the following, I, R, and M_0 denote the *corresponding preprocessed representations*. Simulated prompt locations, prompt-history maps, input patches, and training targets were generated on this common voxel grid.

Prompt and patch representation.
The `MultiClick` models follow an early-prompting strategy, in which user interactions are provided directly at the network input together with the CT image and the current segmentation mask. This makes the interaction information available from the beginning of feature extraction and allows the network to condition its representation on the user input throughout the network. Following the early-prompting strategy of `nnInteractive` [20], point coordinates therefore have to be encoded as spatial guidance maps aligned with the image grid.

Previous interactive-segmentation approaches have used local disks, Gaussian heatmaps, and distance-based representations for this purpose [36, 28]. Spatially localized guidance representations have the advantage that the effect of adding or moving an interaction remains locally limited, whereas global distance-transform representations may change throughout the input domain [36]. Gaussian heatmaps additionally provide a soft representation whose values decrease continuously with distance from the interaction center [28].

Each simulated point prompt was represented by a fixed three-dimensional Gaussian kernel. The point-prompt representation was adopted from the `nnInteractive`-based MEVIS inference implementation used in this work to ensure a consistent prompt encoding between training and inference of the `MultiClick` models. This representation slightly differs from the normalized distance-transform representation described for the original `nnInteractive` framework [20].

The kernel was obtained by sampling an isotropic Gaussian function

$$g(q) = \exp\left(-\frac{\|q\|_2^2}{2\sigma^2}\right), \qquad q \in \{-3, \dots, 3\}^3,$$

with $\sigma = 1.0667$ voxels. The sampled values were subsequently min-max normalized to the interval $[0, 1]$, yielding

$$G(q) = \frac{g(q) - \min_r g(r)}{\max_r g(r) - \min_r g(r)}, \qquad q \in \{-3, \dots, 3\}^3.$$

Thus, $G(0, 0, 0) = 1$, $G(\pm 3, \pm 3, \pm 3) = 0$, and the values decreased with increasing Euclidean distance from the kernel center. The kernel size and standard deviation were retained from the MEVIS `nnInteractive`-based prompt representation and were not optimized separately in this work. For a prompt location $p \in \Omega$, its spatial representation

was defined as

$$B(p) : \Omega \to [0,1], \qquad \text{with}$$
$$B(p)(\omega) = \begin{cases} G(\omega - p), & \omega - p \in \{-3, \dots, 3\}^3, \\ 0, & \text{otherwise.} \end{cases}$$

The resulting $7 \times 7 \times 7$ kernel was centered at the prompt location p. The smoothly decreasing values encode the interaction location together with a small spatial neighborhood while assigning the highest value to the actual prompt position.

Centering the patch at the interaction point ensures that the targeted correction region is contained in the model input while retaining surrounding image context and relevant parts of the current lesion segmentation. For any spatial image, mask, prompt-history map, or tuple of spatially aligned inputs Z, the operation

$$\mathrm{Crop}(Z, p)$$

extracts a spatial patch centered at the prompt location p. A fixed patch size of $128 \times 128 \times 128$ voxels was used to provide a uniform local input extent for all training samples and to match the processing size used for the `MultiClick` network. The fixed patch size provided local context around the interaction while keeping the computational requirements of three-dimensional training manageable. In cases where the requested crop extended beyond the image domain, zero padding was applied to preserve the fixed patch dimension without introducing artificial image or mask structures outside the original image domain. The same padding was applied to all input channels and training targets to preserve their voxel-wise spatial alignment.

Generation of refinement-interaction sequences.
Independently of the point p_{OC} used to generate M_0, a separate simulated initial include prompt was derived from the foreground of the resulting `OneClick` segmentation. This separates the prompt used for generating the initial segmentation from the interaction representation used for subsequent `MultiClick` training. To obtain an initial interaction point that is located within the current segmentation foreground, p_{init} was selected as a maximizer of the Euclidean distance transform of $\mathcal{M}_0$:

$$p_{\mathrm{init}} \in \arg\max_{p \in \mathcal{M}_0} d_{\mathrm{grid}}(p, \Omega \setminus \mathcal{M}_0).$$

The same p_{init} was used for the StepWise and FinalTarget variants. Consequently, the initial-segmentation samples of both variants had identical inputs and differed only in their supervised training target.

The `OneClick` model produced a non-empty initial segmentation mask for every lesion included in the training-data generation. Therefore, p_{init} was well-defined for all lesion cases. This point was used to initialize the include-prompt history, whereas the exclude-prompt history was initialized as zero:

$$P_0^+ = B(p_{\mathrm{init}}) \qquad \text{and} \qquad P_0^- = \mathbf{0}_\Omega.$$

Starting from the initial foreground set $\mathcal{M}_0$, a sequence of up to $T_{\max}$ simulated refinement interactions was generated.

For each interaction step $t \in \{1, \ldots, T_{\max}\}$, the include and exclude error regions were calculated as

$$\mathcal{E}_t^+ = \mathcal{R} \setminus \mathcal{M}_{t-1},$$
$$\mathcal{E}_t^- = \mathcal{M}_{t-1} \setminus \mathcal{R}.$$

The set $\mathcal{E}_t^+$ contains regions that must be added to the current segmentation, whereas $\mathcal{E}_t^-$ contains regions that must be removed. Both error regions were decomposed into sets of 26-connected error components using the convention defined in Section 2.

At each interaction step, the largest remaining error component was selected. This selection strategy provided a deterministic interaction order and prioritized the volumetrically largest residual segmentation error. This represents a heuristic rather than an assumption about the order in which a real user would perform corrections.

To obtain an adequate point representation from the selected component, the prompt was placed at a maximally interior voxel according to the Euclidean distance transform defined in Section 2.2. This guarantees that the prompt lies inside the selected error component.

Before adding the new interaction, both prompt-history maps were multiplied by 0.9. Thus, previous interactions remained available to the network but received progressively lower intensity than more recent prompts. This decay factor was retained from the `nnInteractive`-based interaction representation rather than optimized for the `MultiClick` models. Depending on the component type, the prompt was inserted into the corresponding prompt-history map and the current foreground set was updated by adding or removing the selected component. Point prompts assigned to the same interaction channel were combined using the voxel-wise maximum. This allowed multiple interactions to be represented within a single channel while preserving the normalized intensity range of the individual prompt kernels and preventing overlaps from accumulating additively.

Importantly, the updated mask state $\mathcal{M}_t$ was constructed by applying the selected component-wise correction and was not generated by a `MultiClick` model prediction. Thus, all intermediate mask states and subsequent error components were determined before model training and were independent of the behavior of the subsequently trained model. During training-data generation, the maximum number of generated interactions was set to $T_{\max} = 5$. This value was a methodological design choice rather than an optimized interaction limit.

Initial segmentation training samples.
In addition to the refinement-interaction samples, initial-segmentation samples were included in the training dataset. These samples represent the setting in which no initial segmentation mask is available. Therefore, the current input-mask channel was set to the empty mask

$$M_\emptyset = \mathbf{0}_\Omega.$$

The previously defined p_{init} was used as the positive prompt, with

$$P^{+}_{\text{init}} = B(p_{\text{init}}) \qquad \text{and} \qquad P^{-}_{\text{init}} = \mathbf{0}_{\Omega}.$$

Although M_0 was used to determine the prompt location and served as the stepwise target, it was not provided through the input-mask channel.

Input representation.
Each training input was represented as a four-channel tensor with spatial dimensions $128 \times 128 \times 128$. Accordingly,

$$X_t, X_{\text{init}} \in \mathbb{R}^{4\times 128\times 128\times 128}.$$

The four channels corresponded to the CT image patch, the current input-mask patch, and the include- and exclude-prompt-history maps. For the initial-segmentation samples, the four-channel input was given by

$$X_{\text{init}} = \text{Crop}\left(\left(I, M_{\emptyset}, P^{+}_{\text{init}}, P^{-}_{\text{init}}\right), p_{\text{init}}\right).$$

For the interaction-sequence samples, the input channels were defined as

$$X_t = \text{Crop}\left((I, M_{t-1}, P^{+}_{t}, P^{-}_{t}), p_t\right).$$

Training-target definitions.
Each training target was represented as a binary mask patch of the same spatial size. Two target variants were generated for both refinement-interaction samples and initial-segmentation samples.

For a refinement-interaction sample at step t, the stepwise target represented the segmentation obtained after applying the component-wise optimal correction to the current input mask:

$$Y^{\text{step}}_{t} = \text{Crop}\left(M_t, p_t\right).$$

The final target corresponded to the complete reference segmentation:

$$Y^{\text{final}}_{t} = \text{Crop}\left(R, p_t\right).$$

The FinalTarget strategy follows the target principle used by `nnInteractive`, for which the complete reference object is used as the desired segmentation after each interaction. In contrast, the StepWise strategy corrects only the selected error component in the supervised target while retaining the remaining mask state. This allows the effect of a more preservation-oriented target definition to be investigated.

For an initial-segmentation sample, the stepwise target was given by the initial `OneClick` segmentation M_0:

$$Y^{\text{step}}_{\text{init}} = \text{Crop}\left(M_0, p_{\text{init}}\right).$$

The corresponding final target was given by the complete reference segmentation:

$$Y^{\text{final}}_{\text{init}} = \text{Crop}\left(R, p_{\text{init}}\right).$$

For the StepWise strategy, the initial-segmentation target was chosen as M_0 so that the initialization sample represents the `OneClick`-generated starting state, while subsequent refinement samples teach individual component-wise corrections. In contrast, the FinalTarget strategy uses the complete reference segmentation R as the desired output from every interaction state. Thus, the two strategies differ in whether the desired output follows the controlled correction trajectory or directly represents the complete reference segmentation.

The full procedure is summarized in Algorithm 1. Whenever an arg max contained multiple candidates, the first candidate according to the internal component or voxel ordering was selected.

One generated refinement-interaction training sample is illustrated in Figure 18. The example shows the CT image patch, the current input mask, the include- and exclude-prompt-history maps, a stepwise target mask, and a combined visualization of all four input channels and the target mask.

5.2 Training-Data Variants and Model Configurations

Error-component filtering.
The analysis of the evaluation data and the baseline performance of `nnInteractive` indicated that very small error components represented a distinct interaction scenario. In particular, components that were small both in absolute terms and relative to the corresponding lesion occurred frequently, while interactions targeting such components showed limited correction potential. The corresponding analyses are presented in Section 4.1. Note that the filtering criteria were evaluated on the connected error components on the original voxel grids, before spatial preprocessing and training-patch extraction.

To investigate whether excluding these small components from the training data improved refinement performance, an additional filtered training-data variant was created. After generating the interaction sequences, each refinement-interaction sample was associated with the error component C that had induced the corresponding point prompt. For the filtered training-data variant, the sample was retained only if both of the following conditions were satisfied:

$$|C| \geq 3 \text{ and } \frac{|C|}{|\mathcal{R}|} \geq 0.01.$$

Thus, the respective error component had to contain more than 2 voxels and needed to occupy at least 1% of the corresponding reference-lesion foreground. These specific thresholds were methodological design choices and were not selected or optimized based on the evaluation dataset. The unfiltered variant contained all generated refinement-interaction samples. Filtering was applied only to refinement-interaction samples; all initial-segmentation samples were retained in both training-data variants.

Table 7 summarizes the effect of the filtering procedure on the generated error components. The 29 222 samples correspond to the error components selected during the

Algorithm 1 Generation of `MultiClick` training samples (for one lesion)

Input: CT image I, reference foreground set $\mathcal{R}$, initial `OneClick` foreground set $\mathcal{M}_0$, initial simulated include-prompt location p_{init}, maximum number of interactions $T_{\max}$
Output: Collection of training samples S

$p_{\text{init}} \in \arg\max_{p \in \mathcal{M}_0} d_{\text{grid}}(p, \Omega \setminus \mathcal{M}_0)$ ▷ Choose initial include prompt.

$P_0^+ \leftarrow B(p_{\text{init}})$, $P_0^- \leftarrow \mathbf{0}_\Omega$ ▷ Initialize prompt history.
$S \leftarrow [\,]$

$X_0 \leftarrow \text{Crop}\left((I, \mathbf{0}_\Omega, P_0^+, P_0^-), p_{\text{init}}\right)$ ▷ Construct initial-segmentation input.
$Y_0^{\text{step}} \leftarrow \text{Crop}\,(M_0, p_{\text{init}})$
$Y_0^{\text{final}} \leftarrow \text{Crop}\,(R, p_{\text{init}})$ ▷ Construct initial training targets.
Append $\left(S, (X_0, Y_0^{\text{step}}, Y_0^{\text{final}})\right)$ ▷ Store initial-segmentation sample.

for $t = 1, \ldots, T_{\max}$ **do**
    $\mathcal{E}_t^+ \leftarrow \mathcal{R} \setminus \mathcal{M}_{t-1}, \quad \mathcal{E}_t^- \leftarrow \mathcal{M}_{t-1} \setminus \mathcal{R}$ ▷ Determine remaining error regions.
    $\mathcal{C}_t^+ \leftarrow \mathcal{K}_{26}(\mathcal{E}_t^+), \quad \mathcal{C}_t^- \leftarrow \mathcal{K}_{26}(\mathcal{E}_t^-)$ ▷ Extract connected error components.

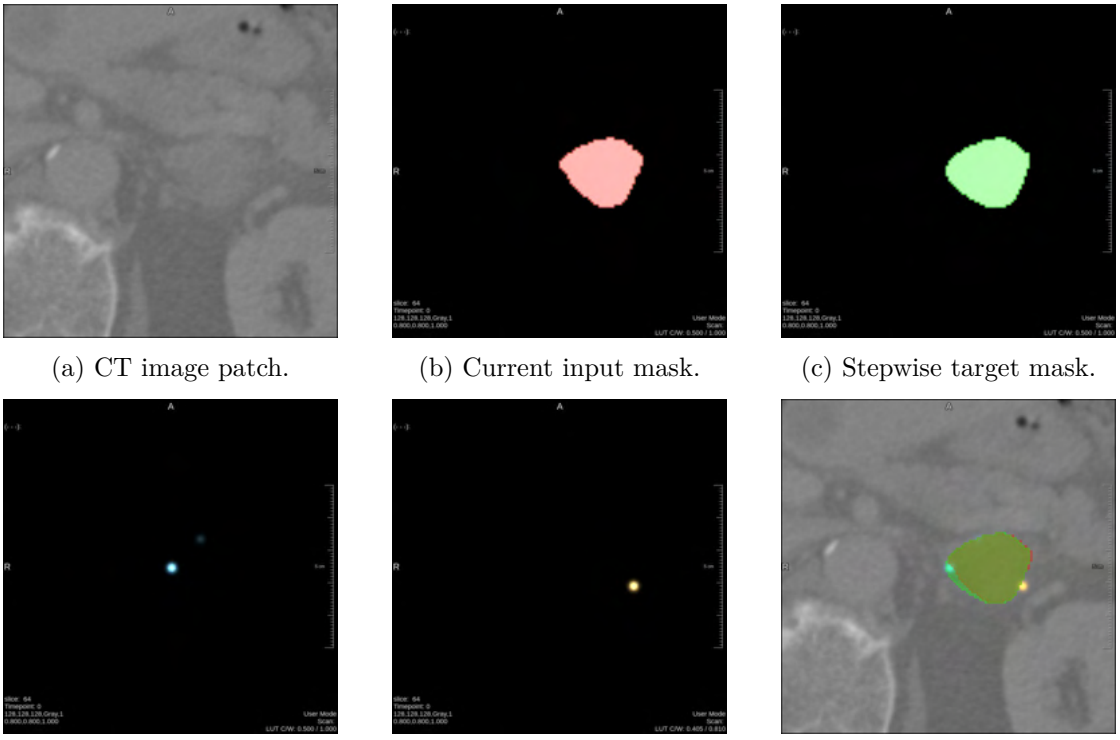
    if $\mathcal{C}_t^+ \cup \mathcal{C}_t^- = \emptyset$ **then**
        break ▷ Stop if no segmentation errors remain.
    end if

    $C_t \in \arg \max_{C \in \mathcal{C}_t^+ \cup \mathcal{C}_t^-} |C|$ ▷ Select largest remaining error component.
    $p_t \in \arg \max_{p \in C_t} d_{\text{grid}}(p, \Omega \setminus C_t)$ ▷ Select maximally interior prompt voxel.

    if $C_t \in \mathcal{C}_t^+$ **then**
        $P_t^+ \leftarrow \max(0.9 P_{t-1}^+, B(p_t))$
        $P_t^- \leftarrow 0.9 P_{t-1}^-$
        $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1} \cup C_t$
    else
        $P_t^+ \leftarrow 0.9 P_{t-1}^+$
        $P_t^- \leftarrow \max(0.9 P_{t-1}^-, B(p_t))$
        $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1} \setminus C_t$
    end if ▷ Update prompt history and current mask.

    $X_t \leftarrow \text{Crop}\left((I, M_{t-1}, P_t^+, P_t^-), p_t\right)$ ▷ Construct refinement input.
    $Y_t^{\text{step}} \leftarrow \text{Crop}\,(M_t, p_t)$
    $Y_t^{\text{final}} \leftarrow \text{Crop}\,(R, p_t)$ ▷ Construct training targets.

    Append $\left(S, \left(X_t, Y_t^{\text{step}}, Y_t^{\text{final}}\right)\right)$ ▷ Store refinement sample.
end for
return S



(a) CT image patch. (b) Current input mask. (c) Stepwise target mask.

(d) Include-prompt history map. (e) Exclude-prompt history map. (f) Combined visualization.

Figure 18: **Example of one generated refinement training sample.** The input consists of a CT image patch, the current segmentation mask, and separate include and exclude prompt-history maps. The shown target corresponds to the stepwise target used for supervised training. **In blue**: Include prompt that represents the current correction interaction. **In orange**: Exclude prompt that originates from a preceding correction step and remains encoded in the prompt history. This example shows only a two-dimensional plane of a three-dimensional sample; thus, not all prompts are shown in the prompt-history map visualization and voxels may appear farther from the component boundary in the displayed axial plane because their three-dimensional distance to the background is limited by the adjacent out-of-plane background.

simulated interaction sequence with at most five interactions per lesion. They therefore represent only a subset of all connected error components identified between the initial and reference masks. Of the 29 222 samples, 17 367 were retained, whereas 11 855 samples were removed. The proportion of removed samples was slightly higher for include-component samples than for exclude-component samples, with 42.0% and 38.9%, respectively.

Table 7: **Error-component filtering statistics.** Components were filtered using a minimum absolute size of three voxels and a minimum relative size of 0.01 with respect to the corresponding lesion.

Component type	Original	Retained	Removed	Removed [%]
Include-component samples	15 503	8 990	6 513	42.0
Exclude-component samples	13 719	8 377	5 342	38.9
All samples	29 222	17 367	11 855	40.6

Resulting model configurations.
Combining the two training-target definitions with the unfiltered and filtered refinement-data variants resulted in four `MultiClick` model configurations. For compact presentation in the following experiments, the models are abbreviated as **SW-A** for StepWise All, **SW-F** for StepWise Filtered, **FT-A** for FinalTarget All, and **FT-F** for FinalTarget Filtered.

Table 8: **Overview of the four `MultiClick` model configurations.**

Model	Abbreviation	Target	Component filtering
StepWise All	SW-A	stepwise target	No
StepWise Filtered	SW-F	stepwise target	Yes
FinalTarget All	FT-A	final target	No
FinalTarget Filtered	FT-F	final target	Yes

5.3 Network Training

All four `MultiClick` models were trained using the same `nnU-Net` architecture and training configuration. The network architecture and optimization settings were kept fixed across all four models to limit differences between the training runs to the investigated training-data variants and stochastic effects of model training. The standard `nnU-Net` training pipeline described in Section 3.1 was used without modification.

nnU-Net plans and network architecture.
The three-dimensional full-resolution configuration was used, with an input patch size of $128 \times 128 \times 128$ voxels.

The resulting architecture was a six-stage three-dimensional PlainConvUNet. The encoder stages used 32, 64, 128, 256, 320, and 320 feature maps, respectively. All convolutional layers used $3\times3\times3$ kernels. The first stage preserved the spatial resolution, whereas subsequent encoder stages performed downsampling with a stride of $2 \times 2 \times 2$. Each encoder and decoder stage contained two convolutional blocks. Each block consisted of a three-dimensional convolution followed by instance normalization and a LeakyReLU activation. Downsampling was performed by strided convolutions, whereas transposed convolutions were used for upsampling in the decoder.

The target spacing stored in the `nnU-Net` plans was $0.8 \times 0.8 \times 1\,\mathrm{mm}$ and therefore matched the spacing metadata of the preprocessed training samples. No additional intensity normalization was configured for any input channel. The CT channel had already been normalized, while the mask and prompt-history map channels were provided in their predefined numerical ranges.

Training procedure.
The four models were trained independently using the default `nnU-Net` v2 trainer. For each model, `nnU-Net` independently generated a random five-fold partition at the level of the individual training cases, and fold 0 was used for model training and validation. This corresponded to a 4:1 training–validation split.

Each model was trained for the complete `nnU-Net` schedule of 1000 epochs with a batch size of two and an initial learning rate of 0.01. The learning rate was gradually reduced throughout training according to the standard `nnU-Net` learning-rate schedule. The training logs were inspected to verify that all training runs used for evaluation completed the full 1000 epochs without premature termination. For subsequent inference, the checkpoint with the best validation performance was selected rather than the checkpoint from the final epoch.

The training objective was the default compound `nnU-Net` loss, combining soft Dice loss and cross-entropy loss. The loss was applied using the standard `nnU-Net` deep supervision strategy.

5.4 Inference

For inference, the MEVIS-internal inference macro in MeVisLab, originally developed for `nnInteractive`, was adapted to the `MultiClick` refinement models. Unless stated otherwise, the same inference procedure was retained, including prompt caching, accumulation of previous interactions, tile-wise processing of inference regions larger than the network patch size, and conversion of the network output into a binary segmentation.

The main adaptations concerned the model-specific preprocessing, network patch size, and input representation. The network input consisted of the same four channels as during training: the normalized CT image patch, the current segmentation-mask patch, and the include and exclude prompt-history maps. In contrast to the `nnInteractive` baseline, the input data were transformed using the preprocessing employed during `MultiClick` training-data generation, described in Section 3.2. The CT image and

current segmentation mask were first transformed to the common `MultiClick` model grid, after which the prompt-history maps were generated on this grid.

As in the `nnInteractive` inference pipeline, inference was performed on a local region centered on the most recently provided point prompt. Its spatial extent was adapted to the accumulated prompt history. Thus, the minimum inference region was $128 \times 128 \times 128$ voxels. Let e_d denote the extent of the accumulated prompt representation along dimension d. The inference-region size was defined as

$$n_d = \max\{128, e_d + 16\}.$$

Restricting inference to a local region provided an input extent consistent with the patch-based training configuration while avoiding unnecessary processing of the complete CT volume.

The model predicted a new segmentation for the current inference region. As in the `nnInteractive` inference pipeline, softmax was applied to the network output to obtain class probabilities, which were subsequently converted into a discrete binary segmentation. The geometry of the local prediction was restored with respect to the original input-image geometry before the segmentation was cached and returned.

This prediction replaced the previously cached segmentation mask; no explicit voxel-wise combination with the preceding mask outside the inference region was performed. It subsequently served as the current segmentation-mask input for the next interaction. Consequently, foreground regions of the preceding segmentation located outside the current inference region were not preserved by the inference pipeline.

5.5 Methodological Limitations

A limitation of the simulated interaction generation is that prompt locations were selected using the Euclidean distance transform on the discrete voxel grid without accounting for physical voxel spacing. For an error component restricted to a single axial slice, each component voxel generally has a background voxel at unit distance in the out-of-plane direction. The maximum distance-transform value may therefore be attained by multiple voxels, potentially including most voxels of the component. In such cases, the implementation selects the first voxel according to the internal voxel ordering. As a result, the selected point may not coincide with the in-plane center of the component and may be located closer to its boundary when considered in the corresponding axial slice, as illustrated in Figure 19. Furthermore, the training data could have been augmented by generating multiple samples for otherwise identical error components using different random prompt locations.

A further limitation concerns the prompt selection for the initial-segmentation training samples of the FinalTarget strategy. The initial prompt p_{init} was derived from the foreground of the `OneClick` segmentation M_0, whereas the desired output of the FinalTarget samples was the reference segmentation R. Since M_0 could differ from R, the selected point was not necessarily a suitable include prompt for the reference lesion. In particular, it could be located close to the reference boundary or even outside the

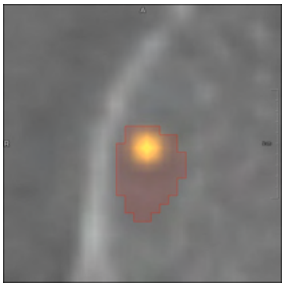


Figure 19: Example of a shifted simulated exclude prompt for a single-slice error component. The component is outlined in red and the prompt representation is shown in orange. Due to multiple voxels attaining the maximum Euclidean distance-transform value, the selected prompt is displaced from the in-plane center of the component.

reference foreground in cases where M_0 contained false-positive regions. This may have introduced inconsistencies between the simulated interaction and the supervised target. Future training-data generation should therefore derive initial include prompts directly from the corresponding training target rather than from M_0.

Finally, one limitation concerns the random training–validation split. The five-fold partition was generated at the level of individual training cases rather than at the lesion or patient level. Since several training samples were generated from different interaction steps of the same lesion, closely related samples could therefore occur in both the training and validation subsets. The validation cases were consequently not fully independent of the training data, which may have resulted in an optimistic estimate of the validation performance used for checkpoint selection. However, the trained models were evaluated in the experiments described in Section 6 using the independent MEVIS-internal evaluation dataset and were therefore not directly affected by this overlap. In addition, the random split was generated independently for each of the four model variants. The exact training and validation subsets were not identical across models, and part of the observed performance differences may therefore be attributable to variation in the sampled training cases rather than exclusively to the investigated target and filtering strategies.

Chapter 6: Experiments

The experimental evaluation investigates `nnInteractive` and the proposed `MultiClick` models in three complementary interaction settings. First, single-prompt segmentation evaluates whether the models can generate a complete lesion segmentation from an initial point prompt. Second, iterative refinement assesses their behavior during successive corrections of an imperfect `OneClick` segmentation. Third, isolated error-component correction evaluates the immediate response to individual correction prompts independently of error propagation across an interaction sequence. Together these experiments assess initial segmentation capability, iterative refinement performance, and the locality and reliability of individual corrections.

6.1 Segmentation Evaluation Metrics

Segmentation quality was evaluated using complementary global and local measures. Dice overlap and spatial boundary distance quantify the overall agreement between the model prediction and the reference segmentation. For interactive refinement, however, global segmentation quality alone does not indicate whether a correction remained confined to the region addressed by the current prompt. Therefore, additional measures were used to quantify preservation and segmentation-quality changes outside the selected error component.

Let M denote a predicted binary segmentation mask, and let R denote the corresponding reference mask. Their foreground voxel sets are denoted by $\mathcal{M} = \mathrm{fg}(M)$ and $\mathcal{R} = \mathrm{fg}(R)$.

Dice.
The Dice similarity coefficient is an overlap-based metric originally introduced for comparing sets and is widely used for the evaluation of medical image segmentations [7, 37]. For the predicted foreground set $\mathcal{M}$ and the reference foreground set $\mathcal{R}$, it is defined as

$$\mathrm{Dice}(\mathcal{M}, \mathcal{R}) = \frac{2|\mathcal{M} \cap \mathcal{R}|}{|\mathcal{M}| + |\mathcal{R}|}.$$

The Dice score takes values in $[0, 1]$. A value of one indicates identical foreground sets, whereas a value of zero indicates that the two foreground sets do not overlap. Empty model predictions were recorded separately and excluded from the calculation of segmentation metrics, since spatial distance metrics are undefined for empty foreground sets.

Hausdorff distance.
The Hausdorff distance is a spatial distance metric for comparing two finite point sets [17]. In medical image segmentation, it can be used to quantify the largest spatial deviation of a predicted segmentation from a reference segmentation [37].

Let $\partial\mathcal{M} \neq \emptyset$ and $\partial\mathcal{R} \neq \emptyset$ denote the boundary sets of the predicted and reference segmentations, respectively. Using the physical point-to-set distance, the directed Hausdorff

distance from the predicted boundary to the reference boundary is defined as

$$h(\partial\mathcal{M}, \partial\mathcal{R}) = \max_{p\in\partial\mathcal{M}} d_{\text{phys}}(p, \partial\mathcal{R}) = \max_{p\in\partial\mathcal{M}} \min_{q\in\partial\mathcal{R}} d_{\text{phys}}(p, q).$$

It therefore corresponds to the largest minimum physical distance from a boundary voxel of the prediction to the reference boundary.

The symmetric Hausdorff distance is then given by

$$\text{HD}(\mathcal{M}, \mathcal{R}) = \max\left(h(\partial\mathcal{M}, \partial\mathcal{R}), h(\partial\mathcal{R}, \partial\mathcal{M})\right).$$

Normalized Hausdorff distance.
The Hausdorff distance is measured in millimeters. A value of zero indicates identical boundary voxel sets, whereas larger values indicate a larger maximum spatial deviation between the two segmentations. Since the maximum is taken over all boundary voxels, the metric is sensitive to individual spatially distant segmentation errors.

To account for differences in lesion size, the Hausdorff distance was normalized by the maximum physical diameter of the corresponding reference lesion. For $\text{diam}_{\text{phys}}(\mathcal{R}) > 0$, the lesion-size-normalized Hausdorff distance is defined as

$$HD_{\text{norm}}(\mathcal{M}, \mathcal{R}) = \frac{HD(\mathcal{M}, \mathcal{R})}{\text{diam}_{\text{phys}}(\mathcal{R})},$$

where the maximum physical diameter of the reference lesion is defined as

$$\text{diam}_{\text{phys}}(\mathcal{R}) = \max_{p,q\in\mathcal{R}} d_{\text{phys}}(p, q).$$

The resulting metric is dimensionless.

Dice-based assessment of local mask preservation.
For interactive refinement, the overall Dice score alone does not indicate whether a correction remained localized to the prompted error region. Let $\mathcal{M}_{t-1} \subseteq \Omega$ denote the foreground voxel set of the segmentation before interaction t, let $\mathcal{M}_t \subseteq \Omega$ denote the resulting model prediction, and let $\mathcal{R} \subseteq \Omega$ denote the reference foreground set. Furthermore, let $\mathcal{C}_t \subseteq \Omega$ be the error component selected for the current interaction. The regions of the masks outside this component are defined as

$$\mathcal{M}_{t-1}^{\text{out}} := \mathcal{M}_{t-1} \setminus \mathcal{C}_t, \qquad \mathcal{M}_t^{\text{out}} := \mathcal{M}_t \setminus \mathcal{C}_{\sqcup},$$

and

$$\mathcal{R}_t^{\text{out}} := \mathcal{R} \setminus \mathcal{C}_t.$$

The same spatial component is therefore excluded from the segmentation before and after the interaction, as well as from the reference mask. The construction of the restricted masks and the resulting Dice-based comparison is illustrated schematically in Figure 20. Although the example shows an include-error component, the same exclusion operation is applied to both include and exclude components. The preservation of the segmentation

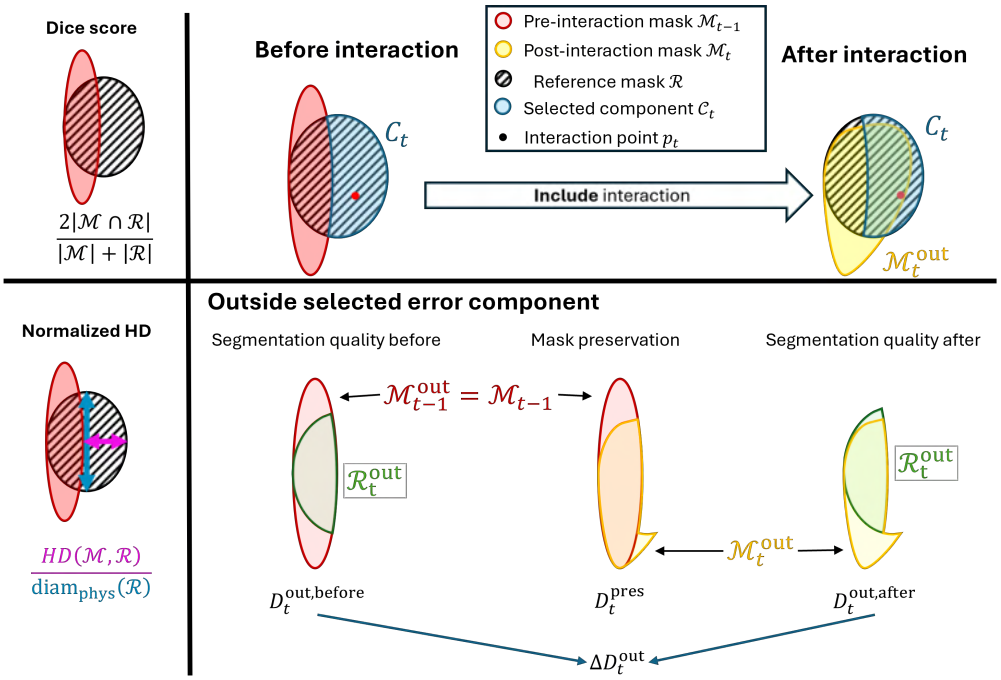


Figure 20: **Schematic illustration of global evaluation metrics, and metrics for mask preservation and segmentation-quality change outside the selected error component.** On the left side is an illustration of the Dice-score and Hausdorff distance. To the right, the Dice-based metrics for restricted assessment outside the selected error component are shown. After excluding $\mathcal{C}_t$ (blue) from the pre-interaction mask (red), post-interaction mask (yellow), and the reference mask, D_t^{pres} measures preservation of non-targeted mask regions. The restricted Dice scores $D_t^{out,before}$ and $D_t^{out,after}$ quantify segmentation quality outside the component before and after the interaction, respectively. Their difference ΔD_t^{out} indicates improvement or deterioration outside the targeted region. Together, the two measures distinguish localized corrections from interactions that unintentionally modify other parts of the segmentation.

outside the selected component is measured by

$$D_t^{\text{pres}} := \text{Dice}\left(\mathcal{M}_{t-1}^{\text{out}}, \mathcal{M}_t^{\text{out}}\right). \tag{5}$$

A value of $D_t^{\text{pres}} = 1$ indicates that the foreground mask outside the selected component remained unchanged. A value below one indicates that foreground voxels were added or removed outside the targeted region.

To assess whether changes outside the selected component improved or worsened the segmentation, the outside masks are additionally compared with the outside reference mask. The Dice scores before and after the interaction are defined as

$$D_t^{\text{out,before}} := \text{Dice}\left(\mathcal{M}_{t-1}^{\text{out}}, \mathcal{R}_t^{\text{out}}\right)$$

and

$$D_t^{\text{out,after}} := \text{Dice}\left(\mathcal{M}_t^{\text{out}}, \mathcal{R}_t^{\text{out}}\right).$$

The resulting change in segmentation quality outside the selected component is given by

$$\Delta D_t^{\text{out}} := D_t^{\text{out,after}} - D_t^{\text{out,before}}. \tag{6}$$

A positive value indicates that additional segmentation errors outside the selected component were corrected, whereas a negative value indicates that the segmentation quality outside the selected region deteriorated.

6.2 Experimental Setups

Three complementary experimental setups were used to evaluate `nnInteractive` and the four proposed `MultiClick` models. For all experiments, we used the evaluation dataset, which is further described in Section 4.1. Only positive and negative point prompts were considered to provide a consistent interaction setting across all models. The setups differ in the segmentation state provided to the model and in whether and how this state is updated after an interaction, as described in Sections 6.2.1–6.2.3. Statistical analyses and visualizations were performed using Python 3.14.0.

6.2.1 Single-Prompt Initial Lesion Segmentation

The first experiment assesses whether an interactive segmentation model can generate an initial lesion segmentation from a single include-point prompt without being provided with a prior lesion mask. The experimental workflow is provided in Figure 21.

The experiment was conducted for all 2494 individual reference lesion masks in the evaluation dataset, including lesions for which no manual correction had been detected in the original annotation workflow. For each lesion, the model received the corresponding CT image and one include-point prompt as input. No initial segmentation mask was provided. The `OneClick` model was not included in this comparison because the reference segmentations in the evaluation dataset were obtained by manually correcting `OneClick`

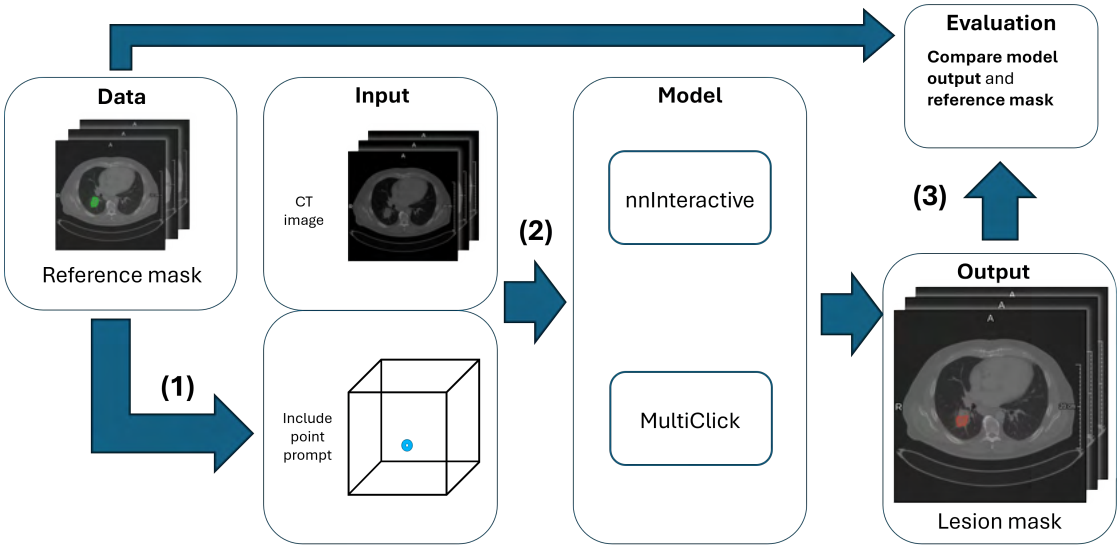


Figure 21: **Experimental setup for single-prompt initial lesion segmentation.** **(1)** A positive point prompt is selected within the reference lesion and **(2)** provided together with the CT image, an empty previous-mask channel, and an empty negative-prompt channel. `nnInteractive` and the four `MultiClick` models independently predict a complete three-dimensional lesion mask, which is **(3)** subsequently compared with the reference segmentation. The setup therefore evaluates the ability of each model to generate a lesion segmentation from a single spatial prompt without information from a previous mask.

predictions. In particular, for lesions without a recorded manual correction, the reference mask is identical to the corresponding `OneClick` prediction.

The prompt location was selected as an EDT maximizer within the reference lesion, following the definition in Section 2.2. The reference mask was used only for prompt generation and subsequent evaluation.

Each lesion was processed using a single model inference without subsequent refinement interactions. The resulting predicted lesion mask was compared with the corresponding reference mask using the evaluation metrics described in Section 6.1.

6.2.2 Iterative Error-Guided Five-Click Refinement

The second experiment evaluates the ability of the models to iteratively refine an imperfect initial lesion mask using error-guided point prompts. Only the 248 lesion cases for which a manual correction of the automatic `OneClick` mask was recorded were used. For these cases, the initial and reference masks differed and therefore contained at least one connected error component from which a correction prompt could be generated.

The five-click refinement loop is shown in Figure 22. For each lesion, the original `OneClick` segmentation was used as the initial current mask and provided through the

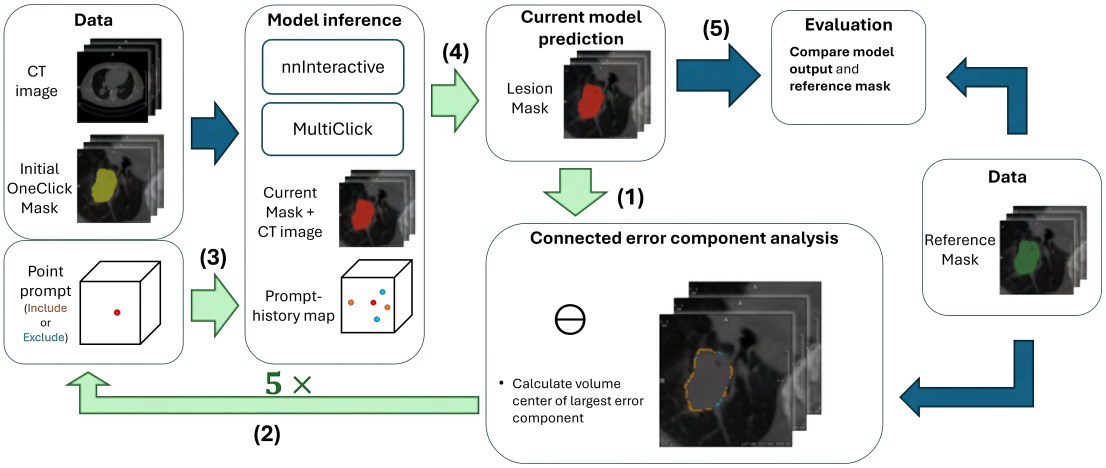


Figure 22: **Experimental setup for iterative error-guided five-click refinement.** **(1)** Connected error components are determined from the differences between the current mask and the reference segmentation. **(2)** The largest remaining error component is selected, and a corresponding include or exclude point prompt is placed at an EDT maximizer within the component and **(3)** added to the accumulated prompt history. From this, together with the CT image and current mask, **(4)** the evaluated model generates an updated lesion segmentation which becomes the new current mask. **(5)** The updated segmentation is evaluated against the reference segmentation. Steps **(1)**–**(5)** were repeated up to five times and the loop was initialized using the initial `OneClick` segmentation. The setup evaluates complete refinement trajectories in which each model response directly determines the segmentation state and the error components available at subsequent interactions.

previous-mask input channel. The connected error components were determined from the voxel-wise difference between the current mask and the corresponding reference mask using 26-connectivity. The error component containing the largest number of voxels was selected, and an EDT maximizer within the selected component was used as the first correction-point prompt. Selecting the largest remaining component provided a deterministic interaction order and prioritized the error region with the largest voxel-wise contribution to the current segmentation error. This selection constitutes a heuristic and is not intended to reproduce the order in which a radiologist would necessarily address individual errors. An include prompt was generated for an include component, whereas an exclude prompt was generated for an exclude component.

All refinement steps were performed within a single iterative inference session. At each iteration, the model received the CT image, the current segmentation mask, and the accumulated click history. After a new correction prompt was added to the interaction history, the model generated an updated lesion mask. This model output was stored as the current mask and reused as the mask input for the next refinement step. All previously generated include and exclude prompts were retained and provided together

with the newly generated prompt.

The updated model output was subsequently compared with the reference mask to identify the largest error component and generate the next correction prompt. This procedure was repeated until a total of five correction prompts had been applied. In the case where a prediction mask became identical to the reference mask before the fifth iteration, no further correction prompt was required. For the analysis of global segmentation performance, the resulting optimal segmentation state was carried forward to the remaining interaction indices. Interaction-based metrics were only evaluated for interactions that were actually performed.

This setup reflects an iterative refinement process in which each model prediction determines the segmentation state available for the next interaction, similar to the training pipeline used for `nnInteractive`. However, it does not isolate the model response to the error components that originally required manual correction by the radiologist, because subsequent error components depend on the preceding model predictions. The isolated error-component experiment complements this evaluation by assessing individual corrections under controlled conditions, using only real corrections by the radiologist.

6.2.3 Isolated Error-Component Correction with Optimal Mask Updates

The third experiment evaluates the model's ability to correct individual error components under controlled conditions. As in the second experiment, only the 248 lesion cases in which manual corrections of the automatic `OneClick` mask occurred were included. However, in contrast to the five-click refinement experiment, the model output was not reused as the current mask for the subsequent correction step. Instead, the selected error component was corrected optimally in the current mask, and the resulting mask was used to initialize the next iteration.

Figure 23 illustrates the workflow of this experiment. For each lesion, the original automatic segmentation was used as the initial current mask. Connected error components were determined from the voxel-wise difference between the current mask and the corresponding reference mask using 26-connectivity. The error component containing the largest number of voxels was selected, and the correction-point location was selected as an EDT maximizer within the selected component. An include prompt was generated for an include error component, whereas an exclude prompt was generated for an exclude error component. The deterministic point placement in this experiment additionally reduced prompt-location variability as a confounding factor when analyzing the influence of component size, shape, and type.

The model received the CT image, the current mask, and the generated correction-point prompt as input and produced a refined lesion-mask prediction. Each correction step was performed in a separate inference session. This was done to isolate the model response to the currently selected error component from the influence of previous interactions.

In parallel, an optimal component-wise correction was applied directly to the current mask. For an include component, all voxels belonging to the selected component were added to the current mask. For an exclude component, all voxels of the selected compo-

nent were removed. The resulting optimally corrected mask was then used as the new current mask for the next iteration, independently of the model prediction.

The model output was recorded and evaluated for each correction step. It was compared with the current input mask, the corresponding reference mask, and the optimally corrected mask. The procedure was repeated while the largest remaining error component contained more than five voxels. This decision was a heuristic restriction of the experimental analysis rather than based on a clinically motivated threshold. The consequences of this restriction are discussed in Section 6.9. Of the 590 error components identified in Chapter 4.1, 413 error components from 230 lesion masks contained at least 6 voxels and were used in this experiment.

Prompt history was not retained because the purpose of this experiment was to isolate the model response to the currently selected error component. Previous prompts would provide additional interaction information unrelated to the current component and could therefore confound the analysis of correction performance with respect to component size, shape, and type.

6.3 Single-prompt Initial Lesion Segmentation

The metrics of `nnInteractive` and the four task-specific models resulting from the single-prompt initial lesion segmentation experiment are summarized in Table 9. All four task-specific models outperformed `nnInteractive` in the single-prompt setting. The `StepWise` variants achieved the strongest overlap-based performance, with `StepWise` Filtered obtaining the highest median Dice score. Both `StepWise` variants also achieved the lowest median Hausdorff distance. In addition, empty predictions occurred almost exclusively for `nnInteractive`. Differences between the two target definitions were smaller than the overall difference between `nnInteractive` and the task-specific models.

Model	Empty outputs	Mean Dice	Median Dice (IQR)	Median HD [mm]
Baseline (`nnInteractive`)	27 (1.1%)	0.591	0.691 (0.411–0.824)	3.000
`StepWise` All	0 (0.0%)	0.809	0.857 (0.759–0.913)	2.500
`StepWise` Filtered	0 (0.0%)	0.819	0.866 (0.769–0.919)	2.500
`FinalTarget` All	2 (0.1%)	0.775	0.832 (0.712–0.900)	2.662
`FinalTarget` Filtered	0 (0.0%)	0.796	0.840 (0.738–0.901)	2.624

Table 9: **Performance of `nnInteractive` and the four task-specific refinement models in the single-prompt initial lesion segmentation experiment.** Predictions were performed for all 2494 lesion cases. Segmentation metrics are reported for non-empty model outputs. The interquartile range (IQR) is reported using the first and third quartiles. HD denotes the Hausdorff distance.

The corresponding distributions are shown in Figure 24. The Dice-score distributions of all four task-specific models were shifted towards higher values and showed smaller interquartile ranges than that of `nnInteractive`. The differences in Hausdorff distance

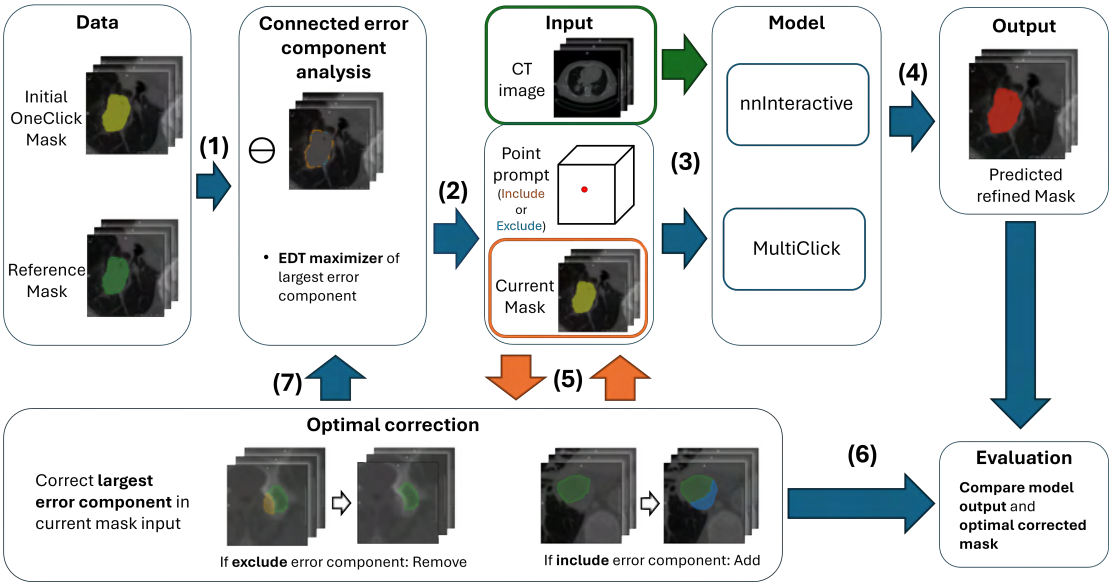


Figure 23: **Experimental setup for isolated error-component correction with optimal mask updates.** **(1)** Starting from the initial `OneClick` mask, connected error components are determined relative to the reference mask. **(2)** The largest remaining component is selected, and an include or exclude point prompt is placed at the EDT maximizer within this component. **(3)** The CT image, current mask, and current point prompt are provided to the evaluated model. **(4)** The resulting model prediction is recorded for evaluation but is not used to update the current mask. Instead, **(5)** the selected error component is corrected optimally in the current mask. **(6)** The optimally corrected mask serves as the component-wise reference for evaluating the model response, and **(7)** the updated current mask is used to determine the next remaining error component. Steps **(2)**–**(7)** are repeated until no error component containing more than five voxels remains. Each model interaction is evaluated independently without retaining prompt history, thereby isolating the response to the currently selected error component while preventing model-induced errors from propagating to subsequent corrections.

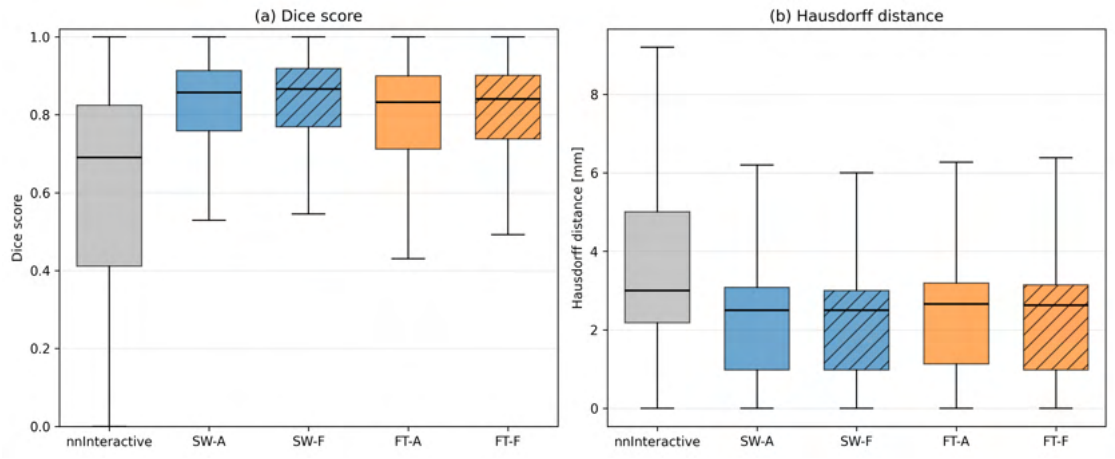


Figure 24: **Distribution of the Dice score and Hausdorff distance for single-prompt initial lesion segmentation across the evaluated models.** Higher Dice scores and lower Hausdorff distances indicate better segmentation performance. The four `MultiClick` models show consistently higher overlap than `nnInteractive`, with the `StepWise` variants achieving the strongest overall performance. Boxplot whiskers indicate the 5th–95th percentile range; outliers are omitted.

were less pronounced, although all task-specific models achieved lower median values than `nnInteractive`.

6.4 Iterative Five-Click Refinement

Overall refinement outcome.
The overall effect of the five-interaction refinement sequence was first assessed by comparing the final segmentation with the common initial `OneClick` segmentation (Figure 25). `SW-F` and `FT-F` achieved the largest improvements, with median Dice-score increases of 0.077 and 0.061, respectively. More than 85% of lesions improved with either filtered variant, while fewer than 10% deteriorated. In contrast, `nnInteractive` showed substantially more heterogeneous outcomes, with approximately equal proportions of improved and worsened lesions. `SW-A` and `FT-A` rarely deteriorated the initial segmentation but frequently showed approximately unchanged Dice scores.

Refinement performance over five interactions.
Figure 26(a)–(b) shows the development of global segmentation performance across the five refinement interactions. All models started from the same initial `OneClick` segmentations, with a median Dice score of 0.767 and a median lesion-size-normalized Hausdorff distance of 0.366. Both `SW-F` and `FT-F` showed the clearest progressive improvement in segmentation overlap, with the largest gains occurring during the earlier interactions and smaller additional improvements thereafter. After five interactions, both filtered variants reached median Dice scores of approximately 0.89. In contrast, `SW-A` and `FT-A`

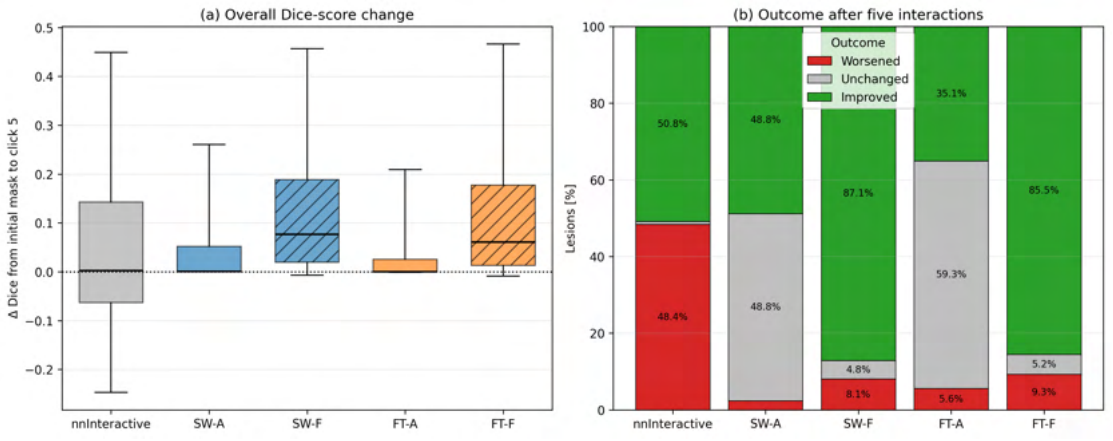


Figure 25: **Per-lesion refinement outcome after five correction interactions.** Panel (a) shows the Dice-score change between the common initial segmentation and the segmentation after five clicks. Positive values indicate improvement, and negative values indicate deterioration. Panel (b) shows the proportion of lesions classified as improved, unchanged, or worsened using an absolute Dice-score change threshold of 0.001. The filtered variants show the most consistently positive final refinement outcomes, whereas `nnInteractive` exhibits greater lesion-wise variability. Boxplot whiskers indicate the 5th–95th percentile range; outliers are omitted.

changed substantially less over the refinement sequence, while `nnInteractive` showed a more heterogeneous response across lesions.

The lesion-size-normalized Hausdorff distance generally decreased during refinement. After five interactions, `FT-F` and `nnInteractive` showed the lowest median normalized Hausdorff distances, followed by `SW-F`, whereas `SW-A` and `FT-A` retained larger boundary deviations. The differing model rankings for Dice score and normalized Hausdorff distance indicate that improvement in segmentation overlap did not directly correspond to the same degree of improvement in boundary agreement.

Preservation of unaffected mask regions.
Panels (c) and (d) in Figure 26 provide a complementary view of the refinement behavior by evaluating changes outside the currently selected error component. The `MultiClick` models preserved these non-targeted regions better than `nnInteractive`. For all four `MultiClick` variants, more than 83% of evaluated interactions achieved a preservation Dice of at least 0.99, compared with 18.2% for `nnInteractive`. `SW-A` and `FT-A` showed the highest preservation, with regions outside the selected component remaining completely unchanged in more than 80% of interactions.

The corresponding Dice-score change outside the selected component further shows whether such modifications improved or deteriorated agreement with the reference mask. The median change was approximately zero for all `MultiClick` variants, whereas `nnInteractive` showed a median decrease of 0.020. Changes outside the targeted component deterio-

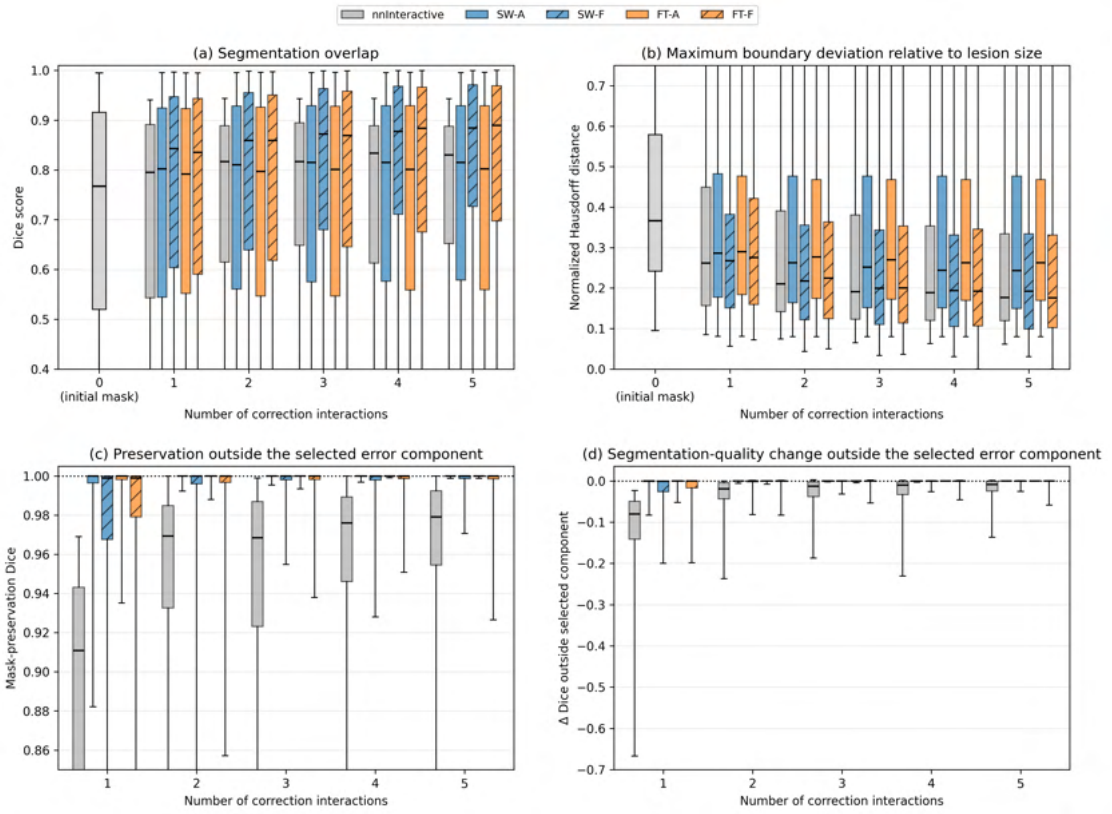


Figure 26: **Refinement performance across five error-guided correction interactions.** Panels (a) and (b) show the distribution of Dice score and lesion-size-normalized Hausdorff distance, respectively. All models start from the same initial `OneClick` model segmentation at interaction 0. Panel (c) shows the preservation Dice between the pre- and post-interaction mask outside the selected error component, whereas panel (d) shows the corresponding change in Dice score relative to the reference mask outside this component. Values in panels (c) and (d) are reported only for interactions that were actually performed. The filtered `MultiClick` variants show the strongest progressive improvement in segmentation overlap, while the unfiltered variants preserve non-targeted mask regions more strictly but respond less strongly to correction prompts. Boxplot whiskers indicate the 5th–95th percentile range; outliers are omitted.

rated the segmentation in 83.5% of nnInteractive interactions. The particularly high preservation of SW-A and FT-A, however, has to be considered together with their limited overall refinement response.

6.5 Isolated Error-Component Correction

Similar to the refinement experiment, the filtered MultiClick model variants showed the strongest correction response. SW-F and FT-F generally shifted the Dice score towards improvement, whereas SW-A and FT-A frequently produced little or no change (Figure 27(a)–(b)). nnInteractive showed a less consistent response, with a tendency towards Dice-score deterioration. The component-wise Oracle achieved considerably larger improvements than any of the evaluated models, indicating that the selected error components generally provided further correction potential that was not fully exploited by the model predictions.

Figure 28 shows that include interactions led to stronger positive correction responses than exclude interactions. The filtered MultiClick variants again exhibited the largest positive Dice-score changes. nnInteractive showed negative median Dice-score changes for both interaction types. The locality analysis further differentiated the model responses (Figure 27(c)–(d)). The MultiClick models generally preserved mask regions outside the selected error component, with the highest preservation observed for SW-A and FT-A. The filtered variants showed larger modifications outside the targeted region, consistent with their stronger overall response to correction prompts. In contrast, nnInteractive produced greater changes outside the selected component. These changes were also more frequently associated with reduced agreement with the reference mask, whereas the Dice score changes outside the targeted region remained concentrated close to zero for the MultiClick models. Overall, the results indicate a trade-off between responsiveness to correction prompts and strict preservation of non-targeted mask regions.

To further examine the variability between individual correction interactions, Figure 29 shows the Dice scores before and after each isolated error-component correction. Interactions are ordered by their pre-interaction Dice score and shown in the same order for all models, allowing model responses to identical correction cases to be compared directly.

The error-component-wise overview confirms the different response patterns observed in the aggregated analysis. SW-A and FT-A frequently produced outputs close to the pre-interaction segmentation, whereas SW-F and FT-F showed larger positive corrections across a broader range of initial Dice scores. nnInteractive exhibited greater variability, including Dice-score reductions for individual interactions as well as Dice-score improvements. Notably, several deteriorating interactions occurred for the same error-component cases, suggesting that particular error-component corrections were consistently difficult. These cases are further investigated in Section 6.6.

The correction potential generally decreased as the pre-interaction Dice score approached one, which was also reflected by the component-wise Oracle (Figure 30(a)). Thus, corrections applied to already highly accurate masks provided only limited poten-

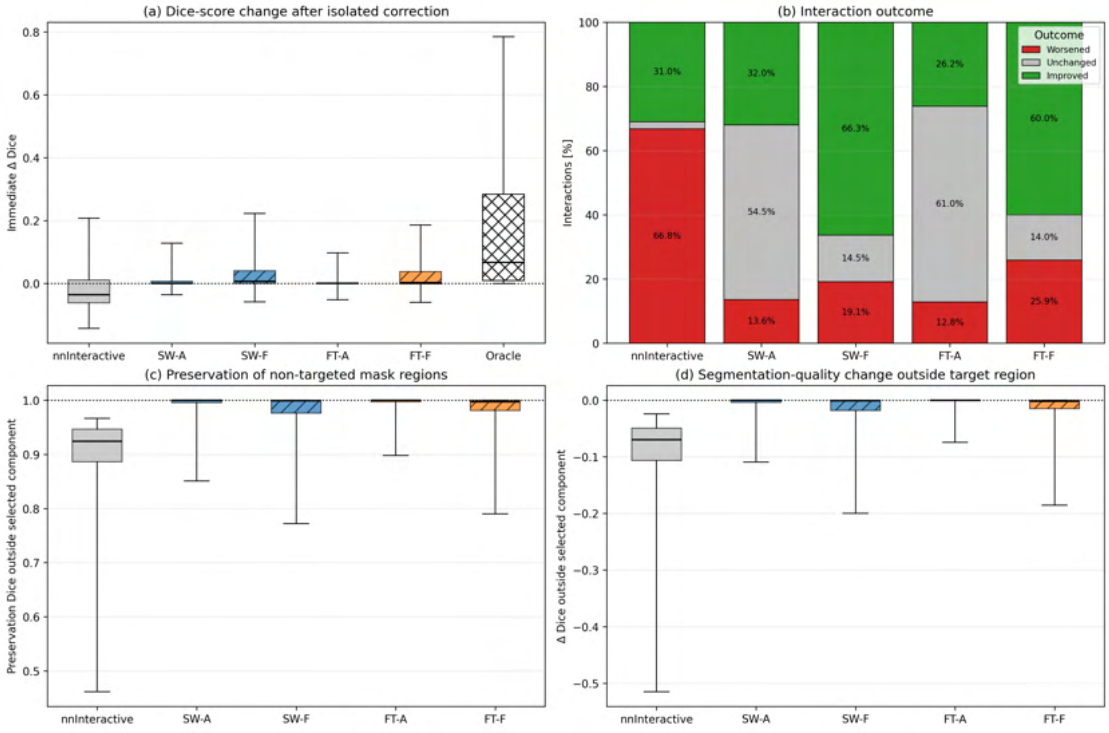


Figure 27: **Performance of isolated error-component corrections across all evaluated interactions.** **(a)** Immediate Dice-score change after correction of a single selected error component, including the component-wise optimal correction (Oracle) as reference. **(b)** Proportion of interactions resulting in improved, unchanged, or worsened Dice scores. Improved and worsened outcomes were defined using an absolute Dice-score change threshold of 0.001. **(c)** Preservation Dice outside the selected error component. **(d)** Change in Dice score relative to the reference mask outside the selected component. Each interaction was evaluated independently using the same pre-interaction mask and selected error component for all models. The filtered `MultiClick` variants show the strongest correction response, whereas the unfiltered variants preserve the existing mask more strictly; `nnInteractive` shows the largest deterioration of non-targeted regions. Boxplot whiskers indicate the 5th–95th percentile range; outliers are omitted.

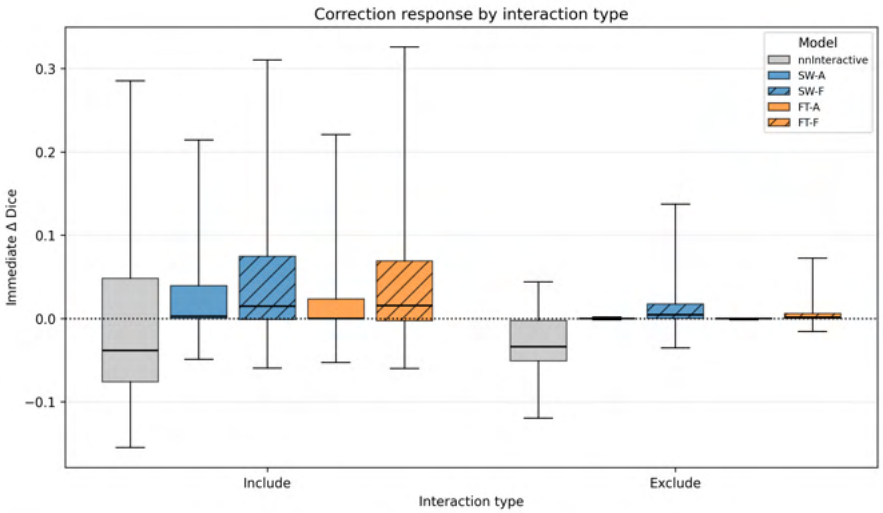


Figure 28: **Dice-score change after isolated error-component correction separated by interaction type.** Boxplots show the immediate Dice-score change for include and exclude interactions for each evaluated model. The figure highlights that include interactions generally produced stronger positive correction responses, whereas exclude interactions more often resulted in little or no global Dice improvements. In particular, the filtered `MultiClick` variants showed the most consistent positive response across both interaction types, while `nnInteractive` tended to reduce the Dice score.

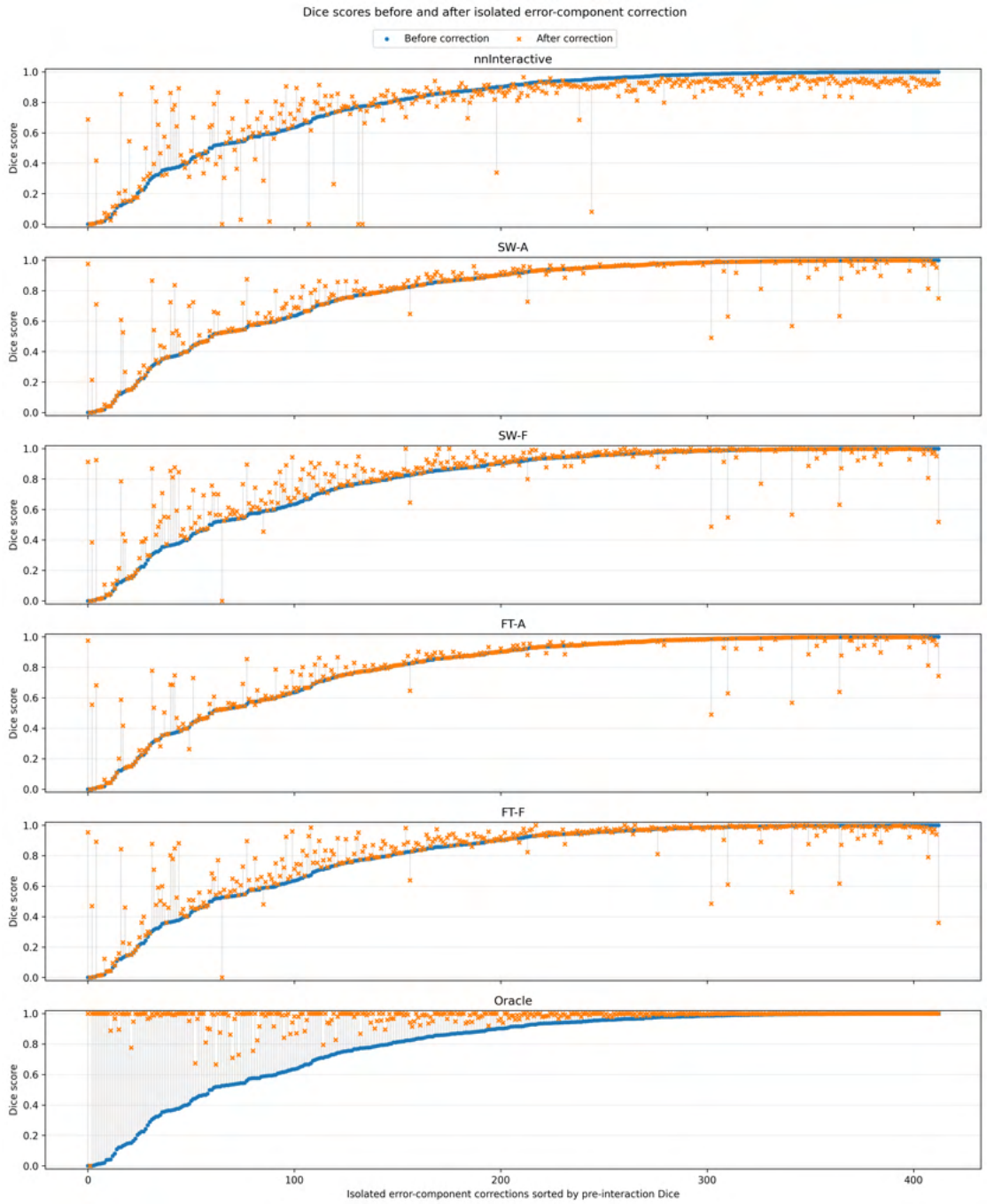


Figure 29: **Dice score before and after all isolated error-component corrections.** Each position on the horizontal axis represents the same error-component correction across all models. Blue markers indicate the segmentation before the correction and orange markers the corresponding model output. The filtered variants produce larger positive corrections across a broader range of cases, while several deteriorating interactions occur at similar positions across models, indicating shared difficult correction cases.

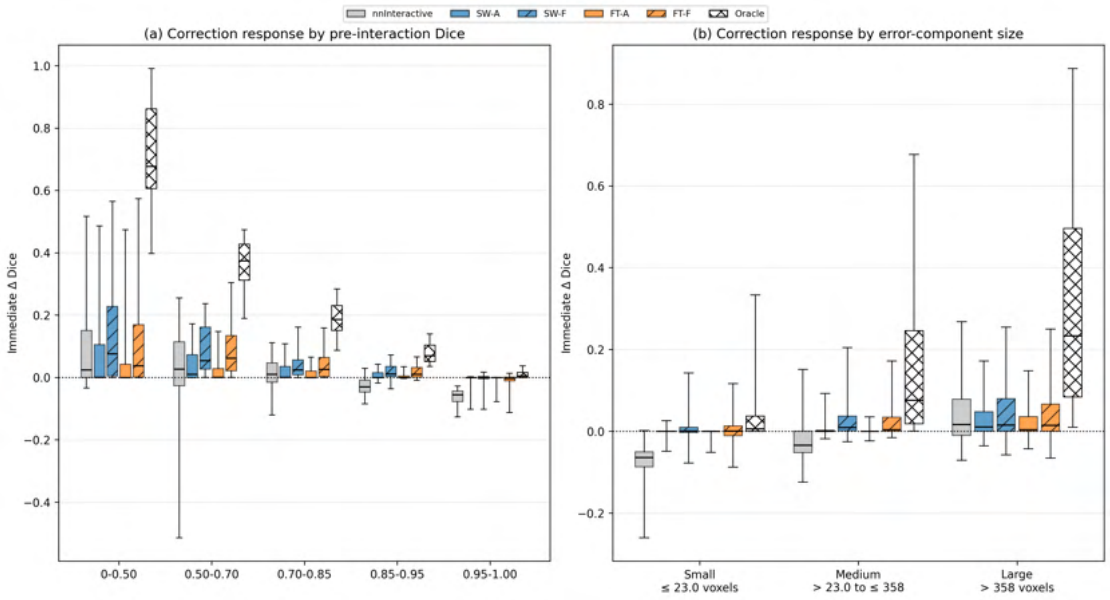


Figure 30: **Influence of pre-interaction segmentation quality and error-component size on isolated correction performance.** Panel (a) shows the immediate Dice-score change for different ranges of pre-interaction Dice score, whereas panel (b) groups interactions according to the size of selected error components. The component-wise Oracle represents the Dice-score improvement obtained when only the selected component is corrected optimally. Both analyses show that the achievable global Dice-score improvement is greatest when the initial segmentation contains substantial remaining error and when the selected component is comparatively large. The filtered `MultiClick` variants exploit this correction potential more strongly than the unfiltered variants.

tial for further global Dice-score improvement. Error-component size showed a complementary effect (Figure 30(b)), with larger components generally providing greater potential for Dice-score improvement than smaller components. The unfiltered `MultiClick` variants frequently remained close to the pre-interaction state.

Since the analysis showed that the shape descriptors, particularly flatness, were associated with component size mainly through single-slice components, the analysis in Figure 31 was restricted to multi-slice error components (see Section 4.1). Within this subset, correction responses were compared across the corresponding quartiles. For both shape descriptors, achievable component-wise Dice improvement increased towards higher ratio values, as indicated by the Oracle distributions. A similar tendency was observed most clearly for `SW-F` and `FT-F`, whereas `SW-A` and `FT-A` remained comparatively close to zero across shape ranges. `nnInteractive` showed a less consistent response and tended to deteriorate the segmentation for components with lower ratio values.

Since the Oracle showed the same general increase across the shape ranges, the ob-

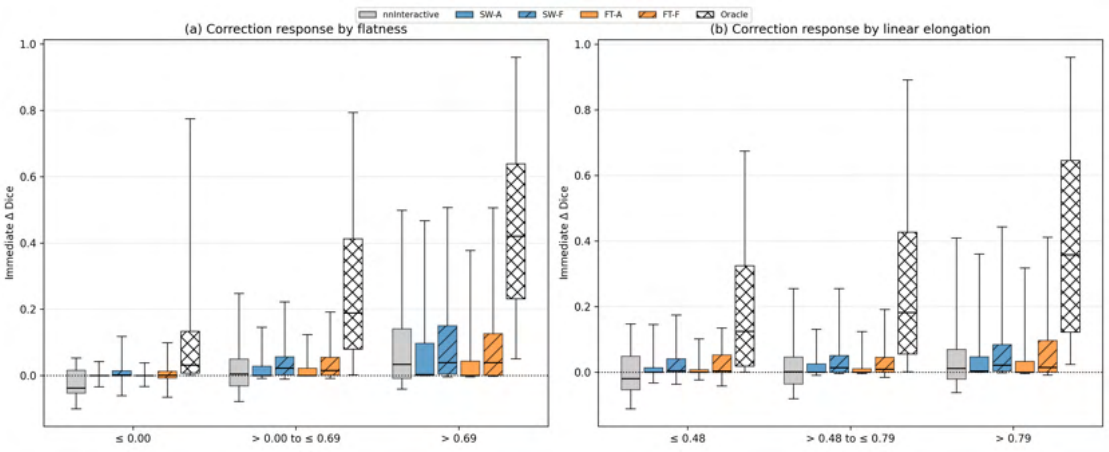


Figure 31: **Influence of PCA-based error-component geometry on isolated correction performance.** Panel (a) groups multi-slice error components according to their flatness ratio, whereas panel (b) groups them according to their linear elongation ratio. The displayed ranges were determined from the first and third quartiles of the corresponding descriptor distributions after excluding single-slice components. The component-wise Oracle shows that the achievable global Dice-score improvement increases towards higher ratio values for both descriptors. A similar tendency is most apparent for the filtered `MultiClick` variants, although the corresponding increase in the Oracle indicates that these differences cannot be attributed solely to a shape-dependent model response. Boxplot whiskers indicate the 5th–95th percentile range; outliers are omitted.

served differences cannot be attributed solely to a shape-dependent model response. Rather, components with higher flatness or elongation ratios provided greater potential for improvement in the global Dice score.

6.6 Representative Results and Failure Cases

The quantitative analyses of the three experiments revealed substantial differences in model behavior as well as recurring failure patterns. To provide a qualitative interpretation of these findings, illustrative successful and unsuccessful segmentation examples were examined. Particular attention was given to correction interactions that resulted in similar deterioration across multiple models, as these may indicate limitations related to the interaction setting rather than isolated model-specific failures.

Illustrative examples of isolated include and exclude corrections are shown in Figure32. The examples visualize different model responses observed. While all four `MultiClick` models largely preserve existing segmentation, their corrections within the selected error component are comparatively small, especially for the unfiltered model variants.

Figure 33 shows five illustrative isolated error-component corrections covering different model-response patterns, including include and exclude corrections and shared failures.

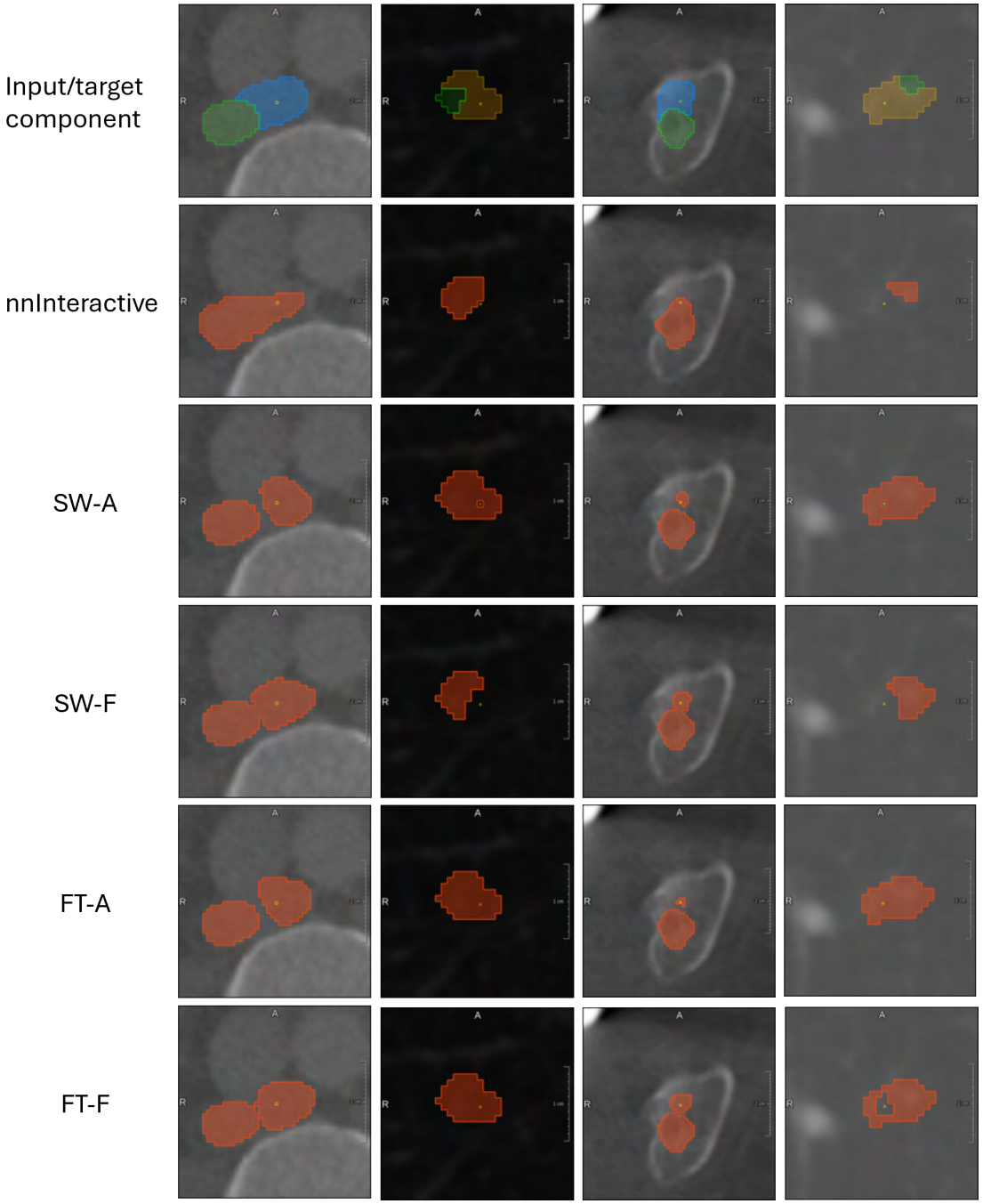


Figure 32: **Representative collection of model outputs for isolated error-component corrections across the evaluated models.** Each column shows one correction case at the same axial image location. The first row depicts the pre-interaction state with the current segmentation mask (green), the selected include (blue) or exclude (orange) error component, and the corresponding point prompt. While all `MultiClick` variants successfully preserve existing mask regions (green), the models tend to perform rather small corrections within the selected error component.

The examples reflect the quantitative findings: filtered `MultiClick` variants generally responded more strongly to correction prompts, whereas the unfiltered variants more often remained close to the pre-interaction mask. `nnInteractive` showed less localized responses in individual cases, with modifications extending beyond the selected error component in some cases.

A subset of interactions resulted in deterioration for all `MultiClick` models. To further specify these deterioration cases, we define a subset of shared failure cases by collecting every correction case in which all `MultiClick` models deteriorated the pre-interaction Dice by more than 0.01. This threshold was chosen to restrict the analysis to cases with a clearly pronounced deterioration of the input segmentation.

Table 10 shows that, for most shared failures, parts of the reference lesion extended beyond the prompt-centered $128 \times 128 \times 128$ voxel inference region. In addition, all shared failures occurred at already high pre-interaction segmentation quality, with Dice scores above 0.92.

	Shared failures	Remaining interactions
Interactions, n	29	384
Lesion outside 128^3 patch	26 (89.7%)	21 (5.5%)
Median pre-interaction Dice	0.999 [0.992, 1.000]	0.888 [0.619, 0.981]

Table 10: **Prompt-centered patch coverage for shared failures and all remaining isolated correction interactions.** For the shared failure cases, the cropped patch does not cover the complete lesion masks in most cases. Furthermore, the overlap before the interaction is already very good in these cases.

Although shared failures were predominantly associated with high pre-interaction Dice scores and incomplete lesion coverage by the prompt-centered inference region, they were not restricted to negligible residual errors. The illustrated exclude correction shown in the fifth column of Figure 33 contains a non-negligible error component for which the component-wise Oracle indicated relevant potential for improvement. Nevertheless, all four `MultiClick` models reduced the Dice score after the correction.

6.7 Comparison of Training-Data Filtering and Target Design

Across both target designs, training-data filtering was associated with greater responsiveness and larger Dice-score improvements in the iterative and isolated correction experiments. The filtered variants improved a larger proportion of lesions and achieved higher final Dice scores, whereas the unfiltered variants frequently produced no measurable change. Although the unfiltered models preserved regions outside the error component more strongly, this apparent stability partly reflected their limited response to the correction prompts.

Differences between the `StepWise` and `FinalTarget` designs were smaller and depended on the evaluation metric. `StepWise` Filtered achieved the strongest Dice-based

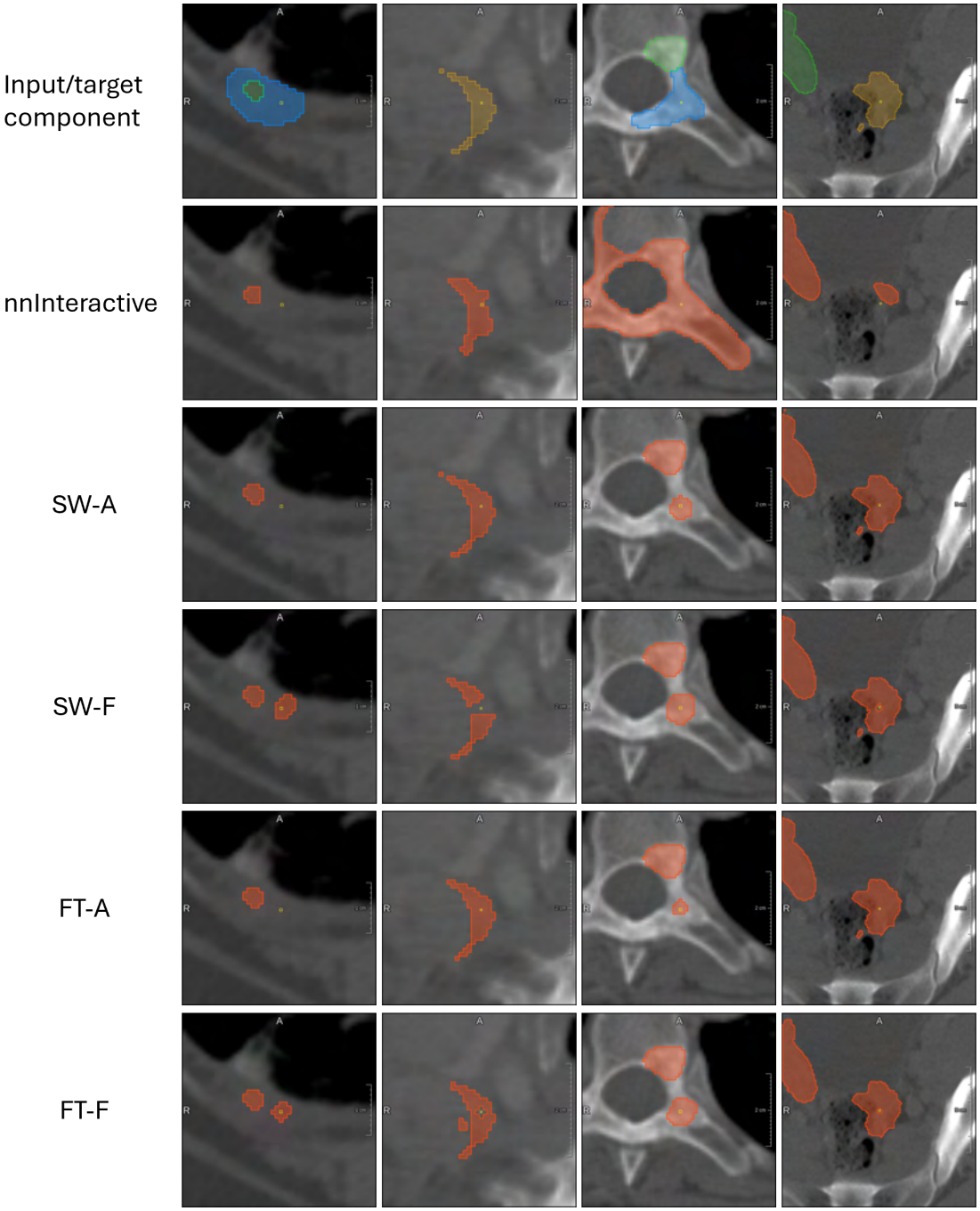


Figure 33: **Illustrative examples of isolated error-component corrections across the evaluated models.** Each column shows one correction case at the same axial image location. The first row depicts the pre-interaction state with the current segmentation mask (green), the selected include (blue) or exclude (orange) error component, and the corresponding point prompt. The examples illustrate different correction behaviors observed in the experiments, including successful corrections, weak or absent responses to the prompt, model-specific differences, and shared failure cases.

results in the single-prompt, iterative-refinement, and isolated correction experiments. `FinalTarget` Filtered remained close in terms of Dice performance and achieved a slightly lower normalized Hausdorff distance and slightly stronger preservation outside the selected error component. Thus, filtering had the more consistent effect on model performance, whereas the target design primarily influenced the balance between segmentation-overlap improvement and spatial preservation.

6.8 Overall Model Comparison

The three experiments provided complementary perspectives on prompt-based lesion segmentation and refinement of volumetric lesion masks. All four `MultiClick` models outperformed `nnInteractive` in the single-prompt segmentation experiment. During iterative refinement, the filtered variants achieved the largest Dice-score improvement and the highest proportions of improved lesion segmentation masks. Although `nnInteractive` achieved the lowest Hausdorff distance, its refinement outcomes showed greater lesion-wise variation and weaker preservation of regions outside the selected error component.

The isolated error-component experiment confirmed that these differences were not caused solely by error propagation along the iterative refinement trajectories. The filtered variants again showed the strongest positive correction response, whereas the unfiltered variants frequently remained close to the pre-interaction mask. Differences between the `StepWise` and `FinalTarget` designs were comparatively small relative to the effect of training-data filtering.

Overall, no model was superior according to every evaluation criterion. However, the filtered `MultiClick` variants provided the best overall balance between responsiveness, segmentation overlap, and preservation of non-targeted mask regions.

6.9 Experimental Limitations

The deterministic point-sampling strategy does not represent the variability of real user interactions and may provide comparatively favorable, centrally located prompts. Furthermore, selecting the largest remaining error component at each interaction is a heuristic and does not necessarily reflect the order in which a user would correct segmentation errors in practice.

In the isolated error-component experiment, model predictions were evaluated independently and optimal component-wise mask updates were used between correction steps. This controlled setup enables individual correction responses to be compared but does not represent a realistic interactive refinement workflow in which previous model outputs and prompt history influence subsequent interactions. Thus, the results do not represent the complete interaction-setting for which the models were designed. They should instead be interpreted as component-wise correction responses conditioned on the current mask and current prompt.

Finally, error components containing fewer than six voxels were excluded from the isolated correction experiment. The reported results therefore do not characterize model

behavior for very small residual segmentation errors.

Chapter 7: Conclusion and Discussion

Aim and approach.
This thesis investigated point-based interactive segmentation and refinement of volumetric lesions in CT images. For this purpose, four task-specific `MultiClick` models were developed using precomputed refinement-interaction samples. The models were designed not only to respond to correction prompts but also to preserve non-targeted regions of an existing lesion segmentation, thereby keeping correction-induced changes localized to the targeted error region. The general-purpose `nnInteractive` model served as a baseline. All five models were evaluated in three complementary settings covering single-prompt initial segmentation, iterative five-click refinement, and isolated error-component correction. Evaluation was performed on a restricted MEVIS-internal dataset containing `OneClick` lesion segmentations and corresponding reference masks obtained after radiological review and manual correction. The dataset was additionally analyzed to characterize the frequency, type, size, and geometry of real radiological mask corrections and thereby provide context for subsequent model evaluation.

Main findings.
Overall, the task-specific `MultiClick` models showed clear advantages over `nnInteractive` for the investigated lesion-segmentation setting. All four `MultiClick` model variants outperformed `nnInteractive` in the single-prompt experiment, with the `StepWise` model variants achieving the strongest overall performance. The stronger initial-segmentation of the `StepWise` variants in comparison to the `FinalTarget` variants performance may partly be explained by the initial prompt construction.

Training-data filtering had a substantially stronger effect on refinement behavior than the choice of training target. Although `StepWise-All` and `FinalTarget-All` showed strong preservation of existing masks, they frequently produced little or no measurable change in response to correction prompts. In contrast, `StepWise-Filtered` and `FinalTarget-Filtered` responded more strongly to the interactions and achieved larger Dice-score improvements in both the iterative and isolated correction experiments. Differences between the `StepWise` and `FinalTarget` variants were comparatively small.

The results of the refinement and isolated error-component experiments further revealed differences in model responses. The `MultiClick` models generally preserved regions outside the targeted error component more strongly than `nnInteractive`, whereas `nnInteractive` more frequently introduced changes outside the prompted region. The isolated error-component experiment showed that the observed differences between the models were not only a consequence of error propagation during iterative refinement, as the filtered `MultiClick` variants also showed the strongest correction response under controlled individual interactions.

Interpretation and discussion.
For interactive refinement of already accurate lesion masks, stability alone is not sufficient. A useful model must balance responsiveness to a correction prompt with preservation of unaffected mask regions. This trade-off became apparent for the `MultiClick`

variants. Their strong preservation of the existing segmentation was accompanied by comparatively weak responses to correction prompts.

At the same time, excessive responsiveness can also be undesirable. When only a small residual error remains, the potential improvement associated with correcting this region is limited. Unintended modifications elsewhere in the mask may outweigh the benefit of the intended correction. Interactive refinement should therefore aim to modify the targeted error region sufficiently while avoiding unnecessary changes to the input segmentation.

These observations also illustrate why global segmentation metrics alone provide an incomplete assessment of interactive refinement. Corrections of small error components may produce only minor changes in the overall Dice score despite being locally correct. Conversely, a model may improve the targeted region while simultaneously deteriorating other parts of the segmentation. Evaluating both the overall segmentation performance and the mask preservation of non-targeted regions provides a more informative characterization of refinement behavior.

Training-data interpretation.
The observed differences between the filtered and unfiltered variants suggest that filtering had a relevant influence on the trained interaction behavior. The complete generated training dataset was strongly dominated by very small and spatially localized error components. For such interactions, the desired target often differs only slightly from the current segmentation mask. Consequently, a training set containing a large proportion of these samples may frequently reinforce predictions that remain close to the existing mask.

This provides one possible explanation for the conservative behavior of the unfiltered variants, which showed strong preservation but frequently produced little or no change to the input segmentation. By removing very small error-component samples, the filtered training variants increased the relative contribution of interactions associated with more recognizable segmentation changes. This may have provided a stronger training signal for responding to the correction prompts and could explain the increased responsiveness observed for `StepWise`-Filtered and `FinalTarget`-Filtered.

At the same time, filtering involves a trade-off. Small errors are a relevant part of interactive refinement, especially for already highly accurate segmentations. Excluding many such samples may improve responsiveness to larger corrections while reducing the representation of fine-scale refinement scenarios during training. The filtering strategy should consequently not be interpreted as showing that small error components are unimportant, but rather that their frequency within the training data may need to be balanced against interactions requiring more substantial mask updates.

Limitations and future work.
Several limitations of the presented approach should be considered when interpreting the results. First, the `MultiClick` models were trained entirely on synthetically generated interaction sequences rather than on correction prompts obtained from real users. The simulated prompts were placed deterministically within selected error components

and did not include perturbations of the prompt location. Consequently, the training distribution represents comparatively idealized interactions and does not account for variability in a real-user scenario. Future work should therefore incorporate more diverse prompt simulation, for example by randomly shifting the point locations within the selected error component. Training or fine-tuning on recorded human interaction sequences would provide a more realistic interaction distribution.

The construction of the initial point prompt in the training data generation resulted in a related limitation. The initial prompt was derived from the corresponding `OneClick` mask M_0 for both training target variants. Because the `FinalTarget` variant used the manually corrected reference mask as the target, the initial prompt may have been positioned less favorably for this target. This could have contributed to the stronger initial-segmentation performance of the `StepWise` variants in comparison to the `FinalTarget` variants. This effect was not investigated separately.

A further limitation concerns the selection of error components during both training-data generation and model evaluation. For `MultiClick` training data, the largest remaining connected error component was selected at each interaction step. The same selection strategy was subsequently used in the iterative and isolated error-component experiments. This creates a favorable interaction setting since larger error components generally provide greater potential for improving global segmentation metrics such as the Dice score. Moreover, the agreement between the interaction policy used for `MultiClick` training and evaluation may favor the proposed models relative to `nnInteractive`, whose original training procedure used a different error-component sampling strategy. Thus, the reported results do not necessarily generalize to interaction sequences in which users select smaller errors or components in a different order. Future work should consider stochastic component selection, randomized interaction order, or sampling strategies that include both small and large residual errors.

The local inference strategy represents another important limitation. The `MultiClick` models were evaluated using a prompt-centered inference region with a minimum size of $128 \times 128 \times 128$ voxels. The resulting prediction replaced the current segmentation mask without merging foreground regions from the preceding mask outside the inference region. Consequently, already correct parts of large lesions could be lost when they extended beyond the local crop, particularly for peripheral correction prompts. The analysis of shared `MultiClick` failures showed that incomplete lesion coverage by the local inference region occurred frequently in such cases. This effect therefore reflects a limitation of the inference pipeline in addition to the learned model behavior. For future implementations, the previous segmentation outside the active inference region could be preserved, the available spatial context could be increased, or an adaptive zooming or region-selection strategy could be applied.

Relatedly, the initial masks used for refinement-training-data generation were produced by the `OneClick` pipeline, which itself operated on a limited local inference region. This may have influenced the spatial characteristics and error patterns of the generated training samples, particularly for larger lesions. Future training-data generation could use larger or adaptive inference regions and should ensure that the complete

lesion context is retained when constructing refinement examples.

The evaluation was additionally restricted to selected error component types. Very small components were excluded from parts of the analysis in order to avoid unstable geometric measurements and corrections dominated by voxel discretization. However, such small errors are relevant for interactive refinement of already accurate masks. Their exclusion therefore limits conclusions about the final stages of refinement. Future experiments should analyze these components separately rather than excluding them entirely, using metrics that are less sensitive to their small absolute size.

The isolated error-component experiment also has limitations related to the interaction sequence. Although the use of optimal mask updates allowed the immediate response to individual correction prompts to be analyzed without propagation of model-induced errors, the resulting input mask still depended on previously corrected error components, while no prompt history was provided to the model. Future work could therefore investigate the influence of error-component ordering more explicitly, either by incorporating the corresponding prompt history or by evaluating each error component independently using the original automatically generated mask as the input segmentation.

Finally, the comparison with `nnInteractive` is not fully symmetric. `nnInteractive` is a general-purpose interactive segmentation model trained with multiple interaction types and a different prompt-generation procedure, whereas the proposed `MultiClick` models were trained specifically for point-based lesion refinement. Restricting all models to point prompts provides a common evaluation setting but does not exploit the full interaction capabilities for which `nnInteractive` was designed. Conversely, the `MultiClick` models benefit from task-specific training data and an interaction policy closely aligned with the evaluation setting. The comparison of the models should therefore be interpreted as a comparison under the point-based lesion-refinement setting investigated in this thesis rather than as a general comparison of interactive segmentation frameworks.

Future work should consequently focus on more realistic and diverse interaction simulation, improved preservation of the existing segmentation during local inference, adaptive inference regions or zooming strategies, and evaluation on additional lesion datasets and interaction policies. A larger and more diverse training dataset would further help determine whether the observed advantages of filtering and target definition generalize beyond the current experimental setting. In addition, the practical benefit of the task-specific models in a clinical workflow remains to be established. Reader studies or human-in-the-loop experiments with radiologists are therefore needed to determine whether the observed improvements translate into practical benefits and to identify evaluation metrics that best reflect clinically relevant refinement performance.

Final conclusion.

This thesis demonstrated that task-specific training can improve point-based interactive lesion segmentation and refinement in CT images compared with a general-purpose interactive segmentation model. The `MultiClick` models showed stronger preservation of existing lesion masks and, for the filtered variants, more effective responses to correction prompts. The results further showed that training-data composition had a greater influence on refinement behavior than the investigated target definition, highlighting

the importance of balancing small preservation-oriented corrections with interactions requiring larger mask updates.

At the same time, the experiments emphasized that interactive refinement cannot be assessed solely by global segmentation accuracy. Effective refinement requires both localized corrections and the preservation of correct areas. While the proposed `MultiClick` models provide a promising task-specific approach for this setting, limitations of the simulated interaction strategy and the local inference pipeline indicate clear directions for further development. Future work should therefore focus on more realistic interaction simulation, improved handling of segmentation regions outside the active inference region, and evaluation across a wider range of datasets and interaction scenarios.

References

[1] Spearman Rank Correlation Coefficient. In *The Concise Encyclopedia of Statistics*, pages 502–505. Springer New York, New York, NY, 2008.

[2] Cvpr 2025–2026: Foundation models for interactive 3d biomedical image segmentation, 2025.

[3] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. FocalClick: Towards Practical Interactive Image Segmentation, April 2022. arXiv:2204.02574 [cs.CV].

[4] Patrick Ferdinand Christ, Mohamed Ezzeldin A. Elshaer, Florian Ettlinger, Sunil Tatavarty, Marc Bickel, Patrick Bilic, Markus Rempfler, Marco Armbruster, Felix Hofmann, Melvin D'Anastasi, Wieland H. Sommer, Seyed-Ahmad Ahmadi, and Bjoern H. Menze. Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields. 2016. Version Number: 1.

[5] M.J.J. De Grauw, E.Th. Scholten, E.J. Smit, M.J.C.M. Rutten, M. Prokop, B. Van Ginneken, and A. Hering. The ULS23 challenge: A baseline model and benchmark dataset for 3D universal lesion segmentation in computed tomography. *Medical Image Analysis*, 102:103525, May 2025.

[6] Andres Diaz-Pinto, Pritesh Mehta, Sachidanand Alle, Muhammad Asad, Richard Brown, Vishwesh Nath, Alvin Ihsani, Michela Antonelli, Daniel Palkovics, Csaba Pinter, Ron Alkalay, Steve Pieper, Holger R. Roth, Daguang Xu, Prerna Dogra, Tom Vercauteren, Andrew Feng, Abood Quraini, Sebastien Ourselin, and M. Jorge Cardoso. DeepEdit: Deep Editable Learning for Interactive Segmentation of 3D Medical Images. volume 13567, pages 11–21. 2022. arXiv:2305.10655 [eess.IV].

[7] Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945.

[8] Yuxin Du, Fan Bai, Tiejun Huang, and Bo Zhao. SegVol: Universal and Interactive Volumetric Medical Image Segmentation, February 2025. arXiv:2311.13385 [cs.CV].

[9] Claude E. Duchon. Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022, August 1979.

[10] E.A. Eisenhauer, P. Therasse, J. Bogaerts, L.H. Schwartz, D. Sargent, R. Ford, J. Dancey, S. Arbuck, S. Gwyther, M. Mooney, L. Rubinstein, L. Shankar, L. Dodd, R. Kaplan, D. Lacombe, and J. Verweij. New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1). *European Journal of Cancer*, 45(2):228–247, January 2009.

[11] Sergios Gatidis, Marcel Früh, Matthias P. Fabritius, Sijing Gu, Konstantin Nikolaou, Christian La Fougère, Jin Ye, Junjun He, Yige Peng, Lei Bi, Jun Ma, Bo Wang, Jia Zhang, Yukun Huang, Lars Heiliger, Zdravko Marinov, Rainer Stiefelhagen, Jan Egger, Jens Kleesiek, Ludovic Sibille, Lei Xiang, Simone Bendazzoli, Mehdi Astaraki, Michael Ingrisch, Clemens C. Cyran, and Thomas Küstner. Results from the autoPET challenge on fully automated lesion segmentation in oncologic PET/CT imaging. *Nature Machine Intelligence*, 6(11):1396–1405, October 2024.

[12] Sergios Gatidis, Felix Peisen, Andreas Wagner, Pauline Ornela Megne Choudja, Ahmed Othman, Antoine Sanner, Nils Grauhan, Suam Kim, Dirk Graafen, Lukas Müller, Tanja LoSSau, Jan Hendrik Moltz, Temke Kohlbrandt, Alessa Hering, Christian La Fougère, Konstantin Nikolaou, and Thomas Küstner. A longitudinal whole-body CT dataset with manually annotated tumor lesions. *Scientific Data*, May 2026.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. arXiv:1512.03385 [cs.CV].

[14] Yufan He, Pengfei Guo, Yucheng Tang, Andriy Myronenko, Vishwesh Nath, Ziyue Xu, Dong Yang, Can Zhao, Benjamin Simon, Mason Belue, Stephanie Harmon, Baris Turkbey, Daguang Xu, and Wenqi Li. VISTA3D: A Unified Segmentation Foundation Model For 3D Medical Imaging, November 2024. arXiv:2406.05285 [cs.CV].

[15] Alessa Hering, Max Westphal, Annika Gerken, Haidara Almansour, Michael Maurer, Benjamin Geisler, Temke Kohlbrandt, Thomas Eigentler, Teresa Amaral, Nikolas Lessmann, Sergios Gatidis, Horst Hahn, Konstantin Nikolaou, Ahmed Othman, Jan Moltz, and Felix Peisen. Improving assessment of lesions in longitudinal CT scans: a bi-institutional reader study on an AI-assisted registration and volumetric segmentation workflow. *International Journal of Computer Assisted Radiology and Surgery*, 19(9):1689–1697, May 2024.

[16] Alessa Denise Hering, Felix Peisen, Teresa Amaral, Sergios Gatidis, Thomas Eigentler, Ahmed Othman, and Jan Hendrik Moltz. Whole-body soft-tissue lesion tracking and segmentation in longitudinal ct imaging studies. *Proceedings of Machine Learning Research*, 2021(Volume 143), 2021. PMLR 143:312-326, 2021.

[17] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850863, September 1993.

[18] Insight Software Consortium. ShapeLabelMapFilter implementation in ITK 5.4.4. `https://github.com/InsightSoftwareConsortium/ITK/blob/v5.4.4/Modules/Filtering/LabelMap/include/itkShapeLabelMapFilter.hxx`. Source code; see `ThreadedProcessLabelObject`; accessed 2026-08-03.

[19] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, February 2021.

[20] Fabian Isensee, Maximilian Rokuss, Lars Krämer, Stefan Dinkelacker, Ashis Ravindran, Florian Stritzke, Benjamin Hamm, Tassilo Wald, Moritz Langenberg, Constantin Ulrich, Jonathan Deissler, Ralf Floca, and Klaus Maier-Hein. nnInteractive: Redefining 3D Promptable Segmentation, 2025. Version Number: 1.

[21] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016.

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, Paris, France, October 2023. IEEE.

[23] Toshiro Kubota, Anna K. Jerebko, Maneesh Dewan, Marcos Salganicoff, and Arun Krishnan. Segmentation of pulmonary nodules of various densities with morphological approaches and convexity models. *Medical Image Analysis*, 15(1):133–154, February 2011.

[24] Thomas Küstner, Felix Peisen, Sergios Gatidis, Andreas Wagner, Ornela Megne, Ahmed Othman, Antoine Sanner, Tanja LoSSau, Jan Hendrik Moltz, Temke Kohlbrandt, and Alessa Hering. Longitudinal-CT, September 2025. Publisher: University of Tübingen.

[25] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, December 2017.

[26] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, January 2024.

[27] Zdravko Marinov, Paul F. Jäger, Jan Egger, Jens Kleesiek, and Rainer Stiefelhagen. Deep Interactive Segmentation of Medical Images: A Systematic Review and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10998–11018, December 2024.

[28] Zdravko Marinov, Rainer Stiefelhagen, and Jens Kleesiek. Guiding the Guidance: A Comparative Analysis of User Guidance Signals for Interactive Segmentation of Volumetric Images. volume 14222, pages 637–647. 2023. arXiv:2303.06942 [cs.CV].

[29] Robert J. McDonald, Kara M. Schwartz, Laurence J. Eckel, Felix E. Diehn, Christopher H. Hunt, Brian J. Bartholmai, Bradley J. Erickson, and David F. Kallmes. The Effects of Changes in Utilization and Technological Advancements of Cross-Sectional Imaging on Radiologist Workload. *Academic Radiology*, 22(9):1191–1198, September 2015.

[30] J. H. Moltz, A. Hering, V. S. Cangalovic, and T. Lossau. Patch-based lesion segmentation from ct, April 2024. Preprint, available via the Universal Lesion Segmentation Challenge 2023.

[31] Yu Qiu and Jing Xu. Delving into Universal Lesion Segmentation: Method, Dataset, and Benchmark. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision ECCV 2022*, volume 13668, pages 485–503. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015. Series Title: Lecture Notes in Computer Science.

[33] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *J. ACM*, 13(4):471494, October 1966.

[34] Tomas Sakinis, Fausto Milletari, Holger Roth, Panagiotis Korfiatis, Petro Kostandy, Kenneth Philbrick, Zeynettin Akkus, Ziyue Xu, Daguang Xu, and Bradley J. Erickson. Interactive segmentation of medical images through fully convolutional neural networks, March 2019. arXiv:1903.08205 [cs.CV].

[35] SimpleITK. Fundamental concepts. `https://simpleitk.readthedocs.io/en/master/fundamentalConcepts.html`.

[36] Konstantin Sofiiuk, Ilia A. Petrov, and Anton Konushin. Reviving Iterative Training with Mask Guidance for Interactive Segmentation, February 2021. arXiv:2102.06583 [cs.CV].

[37] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, December 2015.

[38] Guotai Wang, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sebastien Ourselin, and Tom Vercauteren. DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572, July 2019. arXiv:1707.00652 [cs.CV].

[39] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, Junjun He, and Yu Qiao. SAM-Med3D: Towards General-purpose Segmentation Models for Volumetric Medical Images, 2023. Version Number: 3.

[40] Zhaotao Wu, Jia Wei, Jiabing Wang, and Rui Li. Slice Imputation: Intermediate Slice Interpolation for Anisotropic 3D Medical Image Segmentation, March 2022. arXiv:2203.10773 [eess.IV].

[41] Yuyin Zhou, Qihang Yu, Yan Wang, Lingxi Xie, Wei Shen, Elliot K. Fishman, and Alan L. Yuille. 2D-Based Coarse-to-Fine Approaches for Small Target Segmentation in Abdominal CT Scans. In Le Lu, Xiaosong Wang, Gustavo Carneiro, and Lin Yang, editors, *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, pages 43–67. Springer International Publishing, Cham, 2019. Series Title: Advances in Computer Vision and Pattern Recognition.

[42] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2016*, volume 9901, pages 424–432. Springer International Publishing, Cham, 2016. Series Title: Lecture Notes in Computer Science.