nnU-Net Revisited:
A Call for Rigorous Validation in 3D Medical Image Segmentation

Fabian Isensee[*,1,3], Tassilo Wald[*,1,3,7], Constantin Ulrich[*,1,5,6], Michael Baumgartner[*,1,3,7], Saikat Roy[1], Klaus Maier-Hein[†,1,3,4,5,6,7], Paul Jäger[†,2,3]

[1] Division of Medical Image Computing, German Cancer Research Center (DKFZ), Heidelberg, Germany
[2] Interactive Machine Learning Group (IML), DKFZ, Heidelberg, Germany
[3] Helmholtz Imaging, DKFZ, Heidelberg, Germany.
[4] Pattern Analysis and Learning Group, Department of Radiation Oncology, Heidelberg University Hospital, Heidelberg, Germany
[5] National Center for Tumor Diseases (NCT) Heidelberg, Germany
[6] Medical Faculty Heidelberg, University of Heidelberg, Heidelberg, Germany
[7] Faculty of Mathematics and Computer Science, University of Heidelberg, Heidelberg, Germany
`f.isensee@dkfz-heidelberg.de`

Abstract. The release of nnU-Net marked a paradigm shift in 3D medical image segmentation, demonstrating that a properly configured U-Net architecture could still achieve state-of-the-art results. Despite this, the pursuit of novel architectures, and the respective claims of superior performance over the U-Net baseline, continued. In this study, we demonstrate that many of these recent claims fail to hold up when scrutinized for common validation shortcomings, such as the use of inadequate baselines, insufficient datasets, and neglected computational resources. By meticulously avoiding these pitfalls, we conduct a thorough and comprehensive benchmarking of current segmentation methods including CNN-based, Transformer-based, and Mamba-based approaches. In contrast to current beliefs, we find that the recipe for state-of-the-art performance is 1) employing CNN-based U-Net models, including ResNet and ConvNeXt variants, 2) using the nnU-Net framework, and 3) scaling models to modern hardware resources. These results indicate an ongoing innovation bias towards novel architectures in the field and underscore the need for more stringent validation standards in the quest for scientific progress.

Keywords: Medical Image Segmentation, Validation, Benchmark

---

[*] Equal contribution. Authors are permitted to list their name first in their CVs.
[†] Equal supervision.

1   Introduction

Medical image segmentation remains a highly active area of research, evidenced by the U-Net architecture receiving over 20,000 citations in 2023 alone [29]. The introduction of nnU-Net in 2018 was a pivotal moment, highlighting that careful implementation and configuration of the architecture are more crucial for achieving state-of-the-art results than modifying the architecture itself [23,21]. Despite this, the attraction of innovative architectures from the broader computer vision domain, such as Transformers [33] and Mamba [13], persists. Adaptations of these cutting-edge designs to the medical imaging domain have emerged, with claims of superior performance over the conventional CNN-based U-Net [32,11,15,17,41,38,26,9,12,37,34].

In this paper, we critically examine these claims and find that the current rapid adoption of new methods to the medical domain comes with a lack of stringent validation. As a consequence, we observe that many recent claims of methodological superiority do not hold when systematically tested in a comprehensive benchmark. This trend raises significant concerns, indicating a prevailing attention bias in medical image segmentation towards novel architectures. To overcome this bias and redirect the field towards meaningful methodological progress, we call for a systemic change emphasizing rigorous validation practices. Our study makes the following contributions:

1. We systematically identify validation pitfalls in the field and provide recommendations for how to avoid them.
2. We conduct a large-scale benchmark under a thorough validation protocol to scrutinize the performance of prevalent segmentation methods.
3. Based on this analysis, we identify key methodological components for medical image segmentation as well as a set of suitable benchmarking datasets.
4. We release a series of updated standardized baselines for 3D medical segmentation at https://github.com/MIC-DKFZ/nnUNet. These are based on a residual encoder U-Net within the nnU-Net framework and tailored to accommodate a spectrum of hardware capabilities ("M", "L", "XL").

2   Validation Pitfalls

In the following, we present a collection of predominant validation pitfalls in current practice paired with recommendations on how to avoid them. In Section 4, we underscore the critical need for this initiative by empirically demonstrating how these pitfalls lead to unsupported claims of methodological superiority.

2.1   Baseline-related Pitfalls

Providing a fair and comprehensive comparison to existing work is essential for scientific progress. Currently, we observe a lack of rigour in ensuring meaningful

comparison.

P1: Coupling the claimed innovation with confounding performance boosters: There are multiple ways to artificially boost a method's performance, obfuscating the real impact of the claimed innovation. One example is coupling the claimed innovation with residual connections in the encoder while the baseline uses a vanilla CNN encoder [26]. Another example is coupling the claimed innovation with additional training data not used in baselines [16]. This is even more critical, if the usage of additional data is not made transparent[3]. A related pitfall is to couple the claimed innovation with self-supervised pretraining, while the baselines train from scratch [32]. A third example is coupling the claimed innovation with larger hardware capabilities, i.e. comparing against baselines that are not scaled to the same compute budget (VRAM usage and training time) [31]. Finally, sometimes claimed innovations are based solely on leaderboard results where the method is coupled with 20-fold ensembling, while other leaderboard entries do not use such costly performance boosters[32,16]. **Recommendation (R1):** Meaningful validation entirely isolates the effect of the claimed innovation by ensuring a fair comparison to baselines where the proposed method is not coupled with confounding performance boosters.

P2: Lack of well-configured and standardized baselines: nnU-Net has demonstrated that proper method configuration often impacts performance more significantly than the architecture itself [21]. This suggests that claims of methodological superiority may be misleading if based on comparisons against an ill-configured baseline (i.e. a manually configured U-Net [29] with nontransparent and potentially subpar hyperparameter optimization). Some methods, like nnU-Net, address the "faulty baseline" problem by offering automatic, high-quality, and thus *standardized*, configuration on new datasets. Despite this, many studies continue to claim methodological superiority without benchmarking against any such standardized baseline with a proven high-quality configuration [38,37,34,12,9,11,40]. Beyond auto-configuration frameworks like nnU-Net, it is almost impossible to ensure a high-quality configuration when including existing methods as baselines, because typically no instructions for adaptation to new tasks are provided. This need for manual adjustments, even if equal hyperparameter tuning budget is allocated to all methods, is an error-prone process that ultimately diminishes the relevance of results. **Recommendation (R2):** Beyond the call for ensuring high-quality configuration of baselines, long-term standardization in the field can only be achieved if newly proposed methods are equipped with adaptation instructions, or ideally, are carefully integrated within auto-configuration frameworks to inherit their capabilities.

2.2   Dataset-related Pitfalls

P3: Insufficient quantity and suitability of datasets: The nnU-Net study contains experiments demonstrating 1) the vast diversity of biomedical datasets and 2) the corresponding need of testing on a sufficient number and variety of

datasets when making claims about general methodological advancements [21]. However, the median number of datasets employed in recent studies claiming superior segmentation performance is three [32,16,18,28,17,26,38,20,31,41,37,12,9,11]. Though the number might seem unremarkable on its own, it becomes concerning when considering the varying benchmarking suitability of popular datasets. For instance, as we empirically analyse in Section 4, neither of the two datasets BTCV [25] and BraTS [5,27,6], while being useful environments for solving their respective clinical task, provide a reliable foundation for assessing general methodological advancements. This is due to a high statistical variance (BTCV) and a low systematic variance (BraTS). Despite this, numerous studies claim methodological superiority while at least 50% of the benchmark is made up by either BTCV [37,11,9] or BraTS [14,34,31,28,16]. **Recommendation (R3):** Meaningful validation requires that utilized datasets are a suitable basis for measuring the claimed methodological advancement. This may include sufficient dataset quantity and diversity, as well as benchmarking suitability of individual datasets, as assessed in our study in Section 4.

P4: Inconsistent reporting practices: Standardization of public leaderboard submissions is limited, e.g. allowing varying strategies for ensembling, test time augmentations and post-processing techniques. While perfectly serving the need to demonstrate that a proposed method can push the state-of-the-art when equipped with all bells and whistles, such non-standardized settings undermine the ability to draw meaningful methodological conclusions. Consequently, researchers often resort to custom train/test splits for controlled comparisons against baselines, but these typically involve small test sets that introduce substantial result instability and question the significance of minor performance gains [36,39,41]. Further, the practice of reporting selective results only for specific classes from datasets without justifiable reasons further compromises result integrity [41,11,9,36,39]. **Recommendation (R4):** 5-fold cross-validation with a rotating validation set improves reliability and often represents a pragmatic solution. However, using the same dataset(s) for development and validation bears the risk of implicit overfitting and a lack of generalizability. Thus, ideally, differentiating between a pool of development datasets and an independent pool of test datasets for cross-validation against baselines would offer a more reliable assessment of method performance.

3   Systematic 3D Medical Segmentation Benchmark

Taking these pitfalls and recommendations into account, we revisit recently proposed methods on basis of a systematic and comprehensive benchmark.

3.1   Compared Methods

We categorize methods into CNN-based, Transformer-based, and Mamba-based. **CNN-based:** We include nnU-Net's original configuration using a vanilla U-Net

as well as a variant employing a U-Net with residual connections in the encoder ("nnU-Net ResEnc") which has been part of the official repository since 2019 [22]. In the spirit of avoiding benchmarking of unequal hardware settings in the future (see P1), we introduce new nnU-Net ResEnc presets, which use nnU-Net's existing automatic adaptation of batch and patch sizes to target varying VRAM budgets ("M", "L", "XL"). We further include MedNeXt, a transformer-inspired CNN-modification using ConvNeXt blocks (we test size "L" with kernel sizes "k3" and upkernel "k5") [31], and STU-Net, a series of scaled up U-Nets with increasing parameter counts named "S"(mall), "B"(ase), "L"(arge), and "H"(uge) [20]. **Transformer-based:** We test the SwinUNETR's original version [32] as well as version 2 [17], nnFormer [41], and CoTr, a hybrid architecture combining convolutional and transformer modules [37]. **Mamba-based:** We test the recently proposed U-Mamba model [26] employing Mamba-layers either in the U-Net encoder ("U-Mamba Enc"), or exclusively in the bottleneck ("U-Mamba Bot"). We also include an ablation missing in the original publication using the identical setting while switching off the mamba layers ("No-Mamba Base"). All aforementioned methods were originally implemented in the nnU-Net framework except SwinUNETR(V1+V2), which we integrate into the nnU-Net framework due to incomplete configuration instructions (P2). **Framework comparison:** In addition to comparing recent methods, we also benchmark nnU-Net against a recent alternative framework: Auto3DSeg(Version 1.3.0)[1,28,18,32] is part of the MONAI eco-system [10] and recently created a buzz at MICCAI 2023 by winning several highly competitive challenges like KiTS2023, thereby positioning itself as an alternative to nnU-Net promising the same auto-configuration functionality [1]. The framework is tested by means of three featured architectures ("SegResNet" [28], "DiNTS" [18], "SwinUNETR" [32]).

In the spirit of R1 and R2, we employ a standardized scheme for hyperparameter configuration by either 1) using the self-configuration abilities of methods if available, 2) selecting the configuration closest to the respective dataset if multiple configurations were provided, 3) using the default configuration in case no alternatives were provided or 4) where necessary, decreasing the learning rate until convergence was achieved. All models are trained from scratch. The only exception is SwinUNETR in the Auto3DSeg framework. Altering its default of automatically loading pre-trained weights would have contradicted our hyperparameter configuration scheme. We also employed an equal maximum VRAM budget across all methods by running all trainings on a single NVIDIA A100 with 40GB VRAM. This budget excludes the largest STU-Net variant ("H") from our benchmark.

3.2   Utilized Datasets

Our benchmark utilizes six datasets: BTCV [25], ACDC [7], LiTS [8,4] BraTS2021 [5,27,6] KiTS2023 [19], and AMOS2022 (post challenge Task 2) [24]. We selected datasets based on popularity, allowing us to follow R3 and assess the prevalent datasets w.r.t their suitability for method benchmarking. Given that an effective
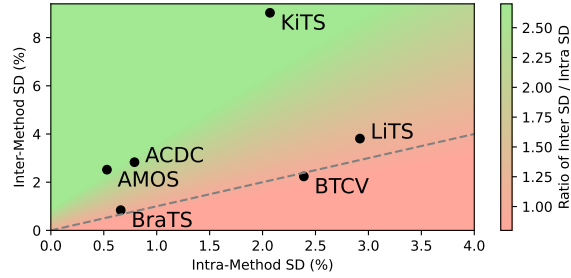
Fig. 1. Benchmarking suitability of popular datasets measured as the ratio of inter-versus intra-method standard deviation (SD). The dashed line denotes a ratio of 1.

benchmarking dataset should enable measuring consistent signals of methodological differences, we derive two requirements for suitability: 1) low standard deviation (SD) of DSC Scores from the same method across the five folds (intra-method SD) indicating statistical stability and a high signal-to-noise ratio. And 2), high SD across different methods (inter-method SD) indicating meaningful signals of methodological differences, i.e. performance does not saturate too fast on the respective task. Our final suitability score is the ratio of inter-method versus intra-method SD.

Following R4, we report results using 5-fold cross-validation, employing splits generated by nnU-Net and applying these consistently across all methods. Since we do not develop new methods in this study, we refrain from distinguishing between development versus test dataset pools. We report results with the average Dice Similarity Coefficients (DSC) being our primary metric, and the Normalized Surface Dice (NSD) as our secondary metric. For both metrics, results are averaged over all classes of each dataset as well as over the five folds to assess generalist segmentation capabilities without delving into problem-specific metric nuances. For datasets featuring hierarchical evaluation regions (BraTS2021, KiTS2023), we calculate metrics for these regions rather than the non-overlapping classes.

4   Results and Discussion

KiTS, AMOS, and ACDC are the most suitable datasets for benchmarking 3D segmentation methods. Fig. 1 shows the outcome of the dataset analysis based on our benchmark (for detailed results see Appendix Table 3). We find that KiTS, AMOS, and ACDC exhibit low statistical noise (intra-method SD) while effectively differentiating between methods, as indicated by a high inter-method SD. Out of the three, KiTS features by far the highest inter-method SD, indicating lowest performance saturation on the task. Conversely, scores on BraTS21 are saturated, with minimal variation both between and within meth-

Table 1. Benchmark results of prevalent 3D medical segmentation methods measured as DSC score [%]. Colors are greener for higher scores, normalized per column. Abbreviations, "RT": Runtime measured in GPU hours on A100 40GB PCIe, "Arch.": Architecture, "TF": Transformer, "Mam": Mamba, "A3DS": Auto3DSeg, "AC": Auto-configuration, "nnU": implemented in nnU-Net framework, "DSC": Dice Similarity Coefficient.

	BTCV n=30	ACDC n=200	LiTS n=131	BraTS n=1251	KiTS n=489	AMOS n=360	VRAM [GB]	RT [h]	Arch.	nnU
nnU-Net (org.) [21]	83.08	91.54	80.09	91.24	86.04	88.64	7.70	9	CNN	Yes
nnU-Net ResEnc M	83.31	91.99	80.75	91.26	86.79	88.77	9.10	12	CNN	Yes
nnU-Net ResEnc L	83.35	91.69	81.60	91.13	88.17	89.41	22.70	35	CNN	Yes
nnU-Net ResEnc XL	83.28	91.48	81.19	91.18	**88.67**	89.68	36.60	66	CNN	Yes
MedNeXt L k3 [31]	84.70	**92.65**	82.14	91.35	88.25	89.62	17.30	68	CNN	Yes
MedNeXt L k5 [31]	**85.04**	92.62	**82.34**	**91.50**	87.74	**89.73**	18.00	233	CNN	Yes
STU-Net S [20]	82.92	91.04	78.50	90.55	84.93	88.08	5.20	10	CNN	Yes
STU-Net B [20]	83.05	91.30	79.19	90.85	86.32	88.46	8.80	15	CNN	Yes
STU-Net L [20]	83.36	91.31	80.31	91.26	85.84	89.34	26.50	51	CNN	Yes
SwinUNETR [32]	78.89	91.29	76.50	90.68	81.27	83.81	13.10	15	TF	Yes
SwinUNETRV2 [17]	80.85	92.01	77.85	90.74	84.14	86.24	13.40	15	TF	Yes
nnFormer [41]	80.86	92.40	77.40	90.22	75.85	81.55	5.70	8	TF	Yes
CoTr [37]	81.95	90.56	79.10	90.73	84.59	88.02	8.20	18	TF	Yes
No-Mamba Base	83.69	91.89	80.57	91.26	85.98	89.04	12.0	24	CNN	Yes
U-Mamba Bot [26]	83.51	91.79	80.40	91.26	86.22	89.13	12.40	24	Mam	Yes
U-Mamba Enc [26]	82.41	91.22	80.27	90.91	86.34	88.38	24.90	47	Mam	Yes
A3DS SegResNet [1,28]	80.69	90.69	79.28	90.79	81.11	87.27	20.00	22	CNN	No
A3DS DiNTS [1,18]	78.18	82.97	69.05	87.75	65.28	82.35	29.20	16	CNN	No
A3DS SwinUNETR [1,32]	76.54	82.68	68.59	89.90	52.82	85.05	34.50	9	TF	No

ods. BTCV exhibits a SD ratio below one, indicating statistical noise may exceed the signal of performance differences between methods. LiTS represents a middle ground in terms of benchmarking suitability. In summary, ACDC, AMOS, and KiTS can be recommended as the most suitable datasets for benchmarking, BraTS, LiTS, and BTCV are observed to be less suitable for this purpose.

CNN-based U-Nets yield best performance. Table 1 shows our experimental results (see Appendix Table 4 for results measured as NSD). CNN-based U-Nets implemented in nnU-Net consistently deliver strong performance across all six datasets. Besides the original nnU-Net, this includes STUNet, ResEnc M/L/XL, MedNeXt and No-Mamba base. MedNeXt consistently stands out with the best performance on all datasets except KiTS, although the gaps are smaller on the datasets with higher benchmarking suitability. Furthermore, MedNeXt's performance gains come at a substantial cost of increased training time (especially k5). Additional experiments in Appendix Table 5 indicate that parts of MedNeXt's advantages can be explained by target spacing selection and are thus not exclusively linked to a superior architecture. Given that STU-Net was primarily introduced with a focus on transfer learning, we analysed the effect of pre-training on the Totalsegmentator dataset in Appendix Table 5

[35]. Importantly, the observed superiority of CNNs is tied to the current experimental setting of training methods from scratch on benchmarks of limited size. While the advantages of Transformers observed in other imaging domains have not yet been realized in medical imaging, the future may hold promise for their success as larger scales of training data become available and transfer learning techniques improve. In contrast to prior claims, Transformer-based architectures (SwinUNETR, nnFormer, CoTr) fail to match the performance of CNNs. This includes not matching performance of the original nnU-Net, which has been released long before the Transformer-based architectures. CoTr shows the best results in the Transformer category, which prior literature related to its convolutional components [30]. U-Mamba initially appears to perform well across segmentation tasks, but comparison against the previously missing baseline "No-Mamba Base" reveals that the mamba layers actually have no effect on performance, and instead, the originally reported gains were due to coupling the method with a residual U-Net (see P1). The fact that SegResNet shows best performance among methods implemented in Auto3DSeg underscores that the observed superiority of CNNs is not merely a bias introduced by nnU-Net.

nnU-Net is the state-of-the-art segmentation framework. We find that none of the three methods featured in Auto3DSeg reaches the original nnU-Net baseline ("org.") performance, indicating a substantial disadvantage due to the underlying Auto3DSeg framework. This negative gap occurs despite significantly lower VRAM usage and training time of the nnU-Net baseline. When comparing the two frameworks with an identical method (SwinUNETR), nnU-Net wins in 5 out of 6 datasets. Following an official Auto3DSeg tutorial [2] we improved the results via manual changes in configuration and further increasing its computing budget, but failed to reach competitive performance (see Appendix Table 2). Taken together, while Auto3DSeg can be pushed to produce state-of-the-art results, as evidenced by its recent challenge wins, its out-of-the-box capabilities do not match nnU-Net.

Scaling models is important especially on larger datasets We tested the effect of model scaling based on two methods: nnU-Net Resenc M/L/XL and STU-Net S/B/L. We find that on the more challenging tasks AMOS and KiTS, a significant boost in performance is observed as the compute budget increases. As expected, the "easier" tasks BTCV and BraTS bear less potential for performance gains from model scaling. These findings underscore the importance of size-awareness and dataset-awareness for meaningful method comparison. For instance, evidence for the superiority of a large new segmentation model should not be based on comparison against a much smaller original nnU-Net.

5   Conclusion

Our benchmark reveals a concerning trend in 3D medical image segmentation: most methods introduced in recent years fail to surpass the original nnU-Net

baseline introduced in 2018. This raises the question: How can we steer the field towards genuine progress? In this study, we link the observed shortcomings to a widespread lack of rigor in method validation. To counteract this, we introduce: 1) A systematic collection of validation pitfalls along with recommendations for their avoidance, 2) The release of updated standardized baselines facilitating meaningful method validation, 3) A strategy for measuring the suitability of datasets for method benchmarking.

Beyond these contributions, achieving true and lasting progress in the field requires a cultural shift, where the quality of validation is valued as much as the novelty of network architectures. Making this shift happen will be the responsibility of method developers, users, and reviewers alike.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Auto3dseg. LINK. Accessed: 2024-01-25.
2. Auto3dseg kits23 tutorial. LINK. Accessed: 2024-03-05.
3. Swinunetr comment on additional training data. `https://github.com/Project-MONAI/research-contributions/issues/68`. Accessed: 2024-01-25.
4. M. Antonelli, A. Reinke, S. Bakas, K. Farahani, and et al. The medical segmentation decathlon. *Nature communications*, 2022.
5. U. Baid, S. Ghodasara, S. Mohan, M. Bilello, and et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.
6. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, and et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 2017.
7. O. Bernard, A. Lalande, C. Zotti, Cervenansky, and et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE TMI*, 2018.
8. P. Bilic, P. Christ, H. B. Li, E. Vorontsov, and et al. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 2023.
9. H. Cao, Y. Wang, J. Chen, D. Jiang, and et al. Swin-unet: Unet-like pure transformer for medical image segmentation. In *ECCV*, 2022.
10. M. J. Cardoso, W. Li, R. Brown, et al. Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022.
11. J. Chen, Y. Lu, Q. Yu, X. Luo, and et al. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.
12. Y. Gao, M. Zhou, and D. N. Metaxas. Utnet: a hybrid transformer architecture for medical image segmentation. In *MICCAI 2021*, 2021.
13. A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

14. A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, and et al. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI Brainlesion Workshop*, 2021.
15. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, and et al. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the WACV*, 2022.
16. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, and et al. Unetr: Transformers for 3d medical image segmentation. In *WACV*, 2022.
17. Y. He, V. Nath, D. Yang, Y. Tang, and et al. Swinunetr-v2: Stronger swin transformers with stagewise convolutions for 3d medical image segmentation. In *MICCAI*, 2023.
18. Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu. Dints: Differentiable neural network topology search for 3d medical image segmentation. In *Proceedings of WACV*, 2021.
19. N. Heller, F. Isensee, D. Trofimova, R. Tejpaul, and et al. The kits21 challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase ct, 2023.
20. Z. Huang, H. Wang, Z. Deng, J. Ye, and et al. Stu-net: Scalable and transferable medical image segmentation models empowered by large-scale supervised pre-training. *arXiv preprint arXiv:2304.06716*, 2023.
21. F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.
22. F. Isensee and K. H. Maier-Hein. An attempt at beating the 3d u-net. *arXiv preprint arXiv:1908.02182*, 2019.
23. F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018.
24. Y. Ji, H. Bai, C. Ge, J. Yang, and et al. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. *Advances in Neural Information Processing Systems*, 2022.
25. B. Landman, Z. Xu, J. E. Igelsias, M. Styner, and et al. 2015 miccai multi-atlas labeling beyond the cranial vault workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, 2015.
26. J. Ma, F. Li, and B. Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.
27. B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, and et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE TMI*, 2014.
28. A. Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *BrainLes 2018, Held in Conjunction with MICCAI*, 2019.
29. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
30. S. Roy, G. Koehler, M. Baumgartner, C. Ulrich, J. Petersen, F. Isensee, and K. Maier-Hein. Transformer utilization in medical image segmentation networks. *arXiv preprint arXiv:2304.04225*, 2023.
31. S. Roy, G. Koehler, C. Ulrich, M. Baumgartner, and et al. Mednext: transformer-driven scaling of convnets for medical image segmentation. In *MICCAI*, 2023.
32. Y. Tang, D. Yang, W. Li, H. R. Roth, and et al. Self-supervised pre-training of swin transformers for 3d medical image analysis. In *CVPR*, 2022.
33. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, and et al. Attention is all you need. *NeurIPS*, 2017.
34. W. Wang, C. Chen, M. Ding, J. Li, and et al. Transbts: Multimodal brain tumor segmentation using transformer. In *MICCAI*, 2021.

35. J. Wasserthal, H.-C. Breit, M. Meyer, M. Pradella, d. Hinck, A. W. Sauter, T. Heye, D. T. Boll, J. Cyriac, S. Yang, M. Bach, and M. Segeroth. Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images. *Radiol Artif Intell.*, 2023.

36. Y. Wu, K. Liao, J. Chen, J. Wang, and et al. D-former: A u-shaped dilated transformer for 3d medical image segmentation. *Neural Computing and Applications*, 2023.

37. Y. Xie, J. Zhang, C. Shen, and Y. Xia. Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. 2021.

38. Z. Xing, T. Ye, Y. Yang, G. Liu, and L. Zhu. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. *arXiv preprint arXiv:2401.13560*, 2024.

39. G. Xu, X. Zhang, X. He, and X. Wu. Levit-unet: Make faster encoders with transformer for medical image segmentation. In *PRCV*, 2023.

40. Y. Zhang, H. Liu, and Q. Hu. Transfuse: Fusing transformers and cnns for medical image segmentation. In *MICCAI*, 2021.

41. H.-Y. Zhou, J. Guo, Y. Zhang, L. Yu, and et al. nnformer: Interleaved transformer for volumetric segmentation. *arXiv preprint arXiv:2109.03201*, 2021.

Table 2. Ablation of Auto3DSeg SegResNet[1,28] following tutorial [2] using fold 0 of KiTS. All experiments were executed on A100 GPUs with 40GB of VRAM. Following the authors recommendation to increase compute resources (manual increase of epochs and patch size), A3DS SegResNet yielded significantly improved results, achieving 87.77% while taking approx. 240 GPU hours (80x30h) and a total of 320GB VRAM (8x40GB). Although this result now surpasses the standard nnU-Net (9h, 7GB VRAM; 86.25%) it is still outperformed by nnU-Net ResEnc M (11h, 9GB; 87.91%) and its larger cousins.

Model	GPU hours	VRAM (GPUs × MB)	Epochs	Batch Size	Patch Size	Spacing	KiTS Fold 0 DSC [%]
nnU-Net (org.)	8.88	1×6901	-	2	$128 \times 128 \times 128$	$1 \times 0.78 \times 0.78$	86.25
nnU-Net ResEnc M	11.39	1×8805	-	2	$128 \times 128 \times 128$	$1 \times 0.78 \times 0.78$	87.91
nnU-Net ResEnc L	35.28	1×24223	-	2	$160 \times 224 \times 192$	$1 \times 0.78 \times 0.78$	88.60
A3DS SegResNet	39.72	1×20267	300	2	$144 \times 224 \times 224$	$1 \times 0.78 \times 0.78$	83.73
A3DS SegResNet	61.28	8×20267	300	8×2	$144 \times 224 \times 224$	$1 \times 0.78 \times 0.78$	76.81
A3DS SegResNet	136.64	8×20267	600	8×2	$144 \times 224 \times 224$	$0.78 \times 0.78 \times 0.78$	85.60
A3DS SegResNet	247.44	8×39873	900	8×2	$224 \times 256 \times 256$	$0.78 \times 0.78 \times 0.78$	87.77

Table 3. Not all datasets can be recommended to develop and compare architectures. We report the standard deviation of DSC scores across folds of the same method (**intra** method SD). We compare this to the standard deviation computed over the average DSC scores across all methods on the dataset (**inter** method SD). The greater the ratio $\frac{inter_{SD}}{intra_{SD}}$, the more suitable is a dataset for separating methods. "SD": Standard Deviation

	BTCV	ACDC	LiTS	BraTS2021	KiTS2023	AMOS2022
nnU-Net (org.)	2.6%	0.8%	3.5%	0.62%	2.0%	0.43%
nnU-Net ResEnc M	2.4%	0.62%	2.6%	0.67%	2.2%	0.57%
nnU-Net ResEnc L	2.7%	0.6%	2.4%	0.57%	1.3%	0.59%
nnU-Net ResEnc XL	2.7%	0.51%	2.4%	0.62%	1.2%	0.43%
MedNeXt L k3	2.1%	0.26%	2.3%	0.66%	0.94%	0.43%
MedNeXt L k5	2.0%	0.2%	2.4%	0.59%	1.2%	0.43%
STU-Net S	2.2%	0.6%	3.3%	0.72%	1.7%	0.42%
STU-Net B	2.3%	0.78%	3.6%	1.0%	1.9%	0.52%
STU-Net L	2.6%	0.85%	2.4%	0.62%	2.1%	0.45%
SwinUNETR	2.7%	0.65%	3.1%	0.75%	2.0%	0.44%
SwinUNETRV2	2.1%	0.51%	2.8%	0.55%	1.7%	0.56%
nnFormer	2.1%	0.21%	2.3%	0.52%	4.2%	0.5%
CoTr	2.8%	0.83%	2.8%	0.69%	1.4%	0.64%
No-Mamba Base	1.9%	0.51%	2.9%	0.55%	2.1%	0.32%
U-Mamba Bot	2.3%	0.59%	2.1%	0.71%	2.7%	0.43%
U-Mamba Enc	2.3%	0.47%	1.7%	0.64%	2.2%	0.5%
A3DS SegResNet	3.0%	0.33%	2.7%	0.52%	1.7%	0.48%
A3DS DiNTS	3.0%	2.2%	2.5%	0.79%	5.3%	1.3%
A3DS SwinUNETR	1.8%	3.6%	6.6%	0.69%	1.5%	0.64%
Averages						
Intra Method SD	2.39%	0.79%	2.89%	0.66%	2.07%	0.53%
Inter Method SD	2.24%	2.83%	3.80%	0.84%	9.03%	2.52%
Inter/Intra Ratio	94%	357%	132%	127%	435%	474%
Averages w/o A3DS						
Intra Method SD	2.35%	0.56%	2.66%	0.66%	1.93%	0.48%
Inter Method SD	1.52%	0.57%	1.68%	0.35%	3.14%	2.28%
Inter/Intra Ratio	65%	102%	63%	53%	163%	477%

Table 4. Normalized Surface Dice (NSD) with tolerance 2 mm for all methods and datasets. Reported values are averages over the five-fold cross-validation. NSD was computed using https://github.com/google-deepmind/surface-distance. Relative performance between methods is consistent with the observations based on Dice alone (see Table 1 and Results section).

Architecture	LiTS	BTCV	ACDC	BraTS2021	KiTS2023	AMOS2022
nnU-Net (org.)	78.26	85.53	94.93	93.64	82.91	91.49
nnU-Net ResEnc M	79.96	86.01	95.50	93.71	84.10	91.72
nnU-Net ResEnc L	80.39	86.08	95.11	93.59	85.93	92.35
nnU-Net ResEnc XL	79.64	85.89	94.90	93.61	86.49	92.64
MedNeXt L k3	81.07	87.78	96.07	93.85	86.29	92.72
MedNeXt L k5	81.26	88.18	96.09	94.04	85.67	92.86
STU-Net S	76.20	85.13	94.27	93.26	81.08	90.81
STU-Net B	77.33	85.30	94.59	93.54	83.08	91.28
STU-Net L	78.85	85.81	95.12	93.66	83.02	92.30
SwinUNETR	73.06	79.79	94.12	93.16	75.91	85.13
SwinUNETRV2	75.38	82.52	95.15	93.15	80.11	88.47
nnFormer	74.66	82.29	95.83	93.22	69.43	82.93
CoTr	77.25	84.10	93.74	93.49	80.92	90.75
No-Mamba Base	78.88	86.14	95.26	93.64	83.56	92.08
U-Mamba Bot	78.91	86.40	95.40	93.65	83.27	92.00
U-Mamba Enc	78.60	84.60	94.33	93.21	83.64	91.25
A3DS SegResNet	76.46	82.01	93.88	93.40	75.61	89.85
A3DS DiNTS	62.49	77.30	83.67	90.36	58.74	82.75
A3DS SwinUNETR	61.16	74.59	83.94	92.00	46.37	86.93

Table 5. Ablation of average DSC when using isotropic spacing for the nnU-Net ResEnc L on ACDC and BTCV instead of the default anisotropic spacing. Results indicate that MedNeXt's performance can in part be explained by its target spacing selection and is thus not exclusively linked to the better architecture. "DSC": Dice similarity coefficient.

Dataset	Method	Patch Size	Spacing	Batch Size	DSC
BTCV	nnU-Net ResEnc L	80x256x256	3x0.76x0.76	2	83.35
	nnU-Net ResEnc L (iso)	192x192x192	1x1x1	2	84.01
	MedNeXt L k3	128x128x128	1x1x1	2	84.70
ACDC	nnU-Net ResEnc L	20x256x224	5x1.56x1.56	10	91.69
	nnU-Net ResEnc L (iso)	96x256x256	1x1x1	3	92.64
	MedNeXt L k3	128x128x128	1x1x1	2	92.65
AMOS	nnU-Net ResEnc L	96x224x224	2x0.71x0.71	2	89.40
	nnU-Net ResEnc L (iso)	192x192x192	1x1x1	2	89.60
	MedNeXt L k3	128x128x128	1x1x1	2	89.62

Table 6. Since STU-Net was presented as a model for transfer learning, we fine-tuned a STU-Net L network, that was pre-trained on the totalsegmentator dataset [35] for 4000 epochs, on the other datasets, analogous to the corresponding publication [20]. Fine-tuning on BraTS did not converge using the default fine-tuning learning rate of 0.001.

	BTCV n=30	ACDC n=200	LiTS n=131	BraTS n=1251	KiTS n=489	AMOS n=360	VRAM [GB]	RT [h]	Arch.	nnU
STU-Net L [20]	83.36	91.31	80.31	91.26	85.84	89.34	26.50	51	CNN	Yes
STU-Net L pretrained [20]	84.28	91.53	81.57	0	88.32	89.46	26.50	51*	CNN	Yes

*Fine-tuning runtime only. Pre-training takes about 4 times longer.