Proceedings of CONF-MLA 2025 Symposium: Applied Artificial Intelligence Research
DOI: 10.54254/2755-2721/2025.BJ26386

Hyperparameter Sensitivity Analysis and Optimization for General Mammal Pose Estimation with KITPose

Yilin Chen[1*], Bicheng Li[1]

Beijing-Dublin International College(BDIC), Bing University of Technology, Beijing, China
**Corresponding Author. Email: yilin.chen@ucdconnect.ie*

Abstract. General mammals pose estimation remains challenging due to high inter-species variability and limited labeled data. While the KITPose framework achieves state-of-the-art performance on benchmarks like AP-10K and AnimalPose, its hyperparameter configuration was empirically fixed and not systematically studied. This paper presents the first comprehensive hyperparameter sensitivity analysis and optimization for KITPose. A two-stage experimental design is adopted: Stage 1 conducts single-variable controlled experiments across six key hyperparameters—transformer depth, attention head count, learning rate, optimizer, scheduler, and augmentation scale—on both AP-10K and AnimalPose. Results reveal that depth, head count, and learning rate are the most influential factors. Stage 2 applies Bayesian optimization via Optuna to jointly optimize these three parameters within empirically grounded ranges. Automated tuning confirms that shallow architectures (depth ≤ 2) with multi-head attention (≥ 2 heads) and moderate learning rates (1e-4–4e-4) consistently outperform deeper or under-parameterized variants. Notably, increasing depth to 4 leads to sharp performance degradation (>10% mAP drop), validating the original KITPose design.

Keywords: Hyperparameter Optimization, Animal Pose Estimation, AP10K, AnimalPose

1. Introduction

Animal pose estimation has attracted increasing attention due to its applications in ecology, wildlife monitoring, and veterinary behavior analysis [1]. Unlike human pose estimation, which benefits from abundant annotations and consistent body structure, animal pose estimation is challenged by species diversity, morphological variability, and limited labeled data [2]. These factors hinder the development of robust and generalizable models. For example, the success of human pose estimation is largely based on large-scale datasets, which are lacking for animal pose estimation [3]. The variability of animal species and the need for species-specific labeled datasets further complicate the task [4].

To address these issues, the Keypoint Interactive Transformer for General Mammal Pose Estimation(KITPose) framework was proposed as a unified solution for general mammal pose estimation. It employs a dual-encoder transformer design, body part prompting, and an adaptive weighting mechanism for keypoints [2]. While KITPose achieves state-of-the-art results on benchmarks such as AnimalPose and AP-10K [5], the original work emphasizes architectural

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.BJ26386

innovation and adopts a fixed set of hyperparameters, leaving a gap in understanding how hyperparameter choices affect performance. This is a common issue in the field, as many studies focus on model architecture while neglecting the impact of hyperparameter tuning [4].

This study adopts a quantitative experimental methodology consisting of a two-stage design. In Stage 1, single-variable controlled experiments are conducted on AP-10K and AnimalPose to evaluate the sensitivity of six common hyperparameters—transformer depth, attention head count, learning rate, optimizer type, scheduler type, and augmentation scale—while keeping other settings fixed. In Stage 2, the most influential hyperparameters identified in Stage 1 are further optimized using Optuna-based Bayesian optimization, enabling automated multi-parameter tuning focused on depth, head count, and learning rate.

All experiments are conducted with deterministic settings and a fixed random seed. Adaptive weighting is disabled, and CutMix is modified for consistency. This two-stage design aims to provide practical insights into hyperparameter tuning for transformer-based animal pose models and to highlight its critical role in improving generalization across diverse mammal species.

2. Methodology

2.1. KITPose framework & setup

This study adopts the KITPose framework as the backbone for general mammal keypoint estimation. The model architecture follows the official KITPose implementation, combining a multi-resolution HRNet encoder with a dual-encoder Transformer module. In the baseline setup, the Transformer is configured with a depth of 2 and a single attention head, while the body-part prompting mechanism is enabled with Np = 4N for AP-10K and Np = 5N for AnimalPose. All experiments employ heatmap-based supervision using a 2D Gaussian centered at the ground-truth joint location. Experiments are conducted on two representative datasets: AnimalPose and AP-10K. AnimalPose covers 20 anatomical keypoints across 5 quadruped categories, while AP-10K is a large-scale benchmark consisting of 17 keypoints over 23 mammal species. Both datasets are annotated in COCO-style JSON format and processed into a unified pipeline. All input images are resized to 256×256 resolution, and the output heatmaps are generated at 64×64 with a Gaussian sigma of 2.

To ensure reproducibility, a deterministic training regime is adopted, with the random seed fixed at 22 and cuDNN benchmarking disabled. Additionally, a modified CutMix strategy is introduced in which the mixing region and index are controlled via a fixed RandomState(22) generator. This adjustment reduces augmentation randomness, allowing for more consistent analysis of hyperparameter effects. All models are trained using adaptive loss functions similar to those originally employed in KITPose, in order to control variables and ensure that experimental results are more reliable.

Training is conducted on four NVIDIA RTX 3090 GPUs using PyTorch 1.10.1 with CUDA 11.1. Unless otherwise specified, baseline experiments are trained for 250 epochs with batch sizes of 32 (AnimalPose) and 64 (AP-10K). The optimizer, learning rate, and related training hyperparameters follow the default KITPose configuration, and their individual effects will be further analyzed in the subsequent sensitivity experiments.

DOI: 10.54254/2755-2721/2025.BJ26386

2.2. Stage 1: hyperparameter sensitivity analysis

2.2.1. Hyperparameters study

To assess how individual hyperparameters influence the performance of general mammal pose estimation, a series of controlled experiments based on the KITPose framework were designed. In this stage, one hyperparameter was systematically varied at a time while all other settings were kept fixed. The experiments were conducted under a consistent training regime established in Section 2, with deterministic mode enabled and CutMix modified to reduce randomness. Performance was evaluated using COCO-style metrics, including AP@[0.5:0.95], AP@0.5, AP@0.75, as well as average recall (AR) and size-specific AP scores (AP(M), AP(L), etc.).

According to findings on the impact of common hyperparameters and general guidelines for hyperparameter exploration [6-7], six hyperparameters were selected based on their critical role in the KITPose model architecture and their substantial influence on optimization in transformer-based frameworks.

A. Transformer Depth

The number of self-attention layers in each encoder branch is controlled by this parameter. However, as noted by Chen and Tsou [8], excessive depth may impose computational burdens rather than providing representational benefits, particularly in convolutional neural networks. To evaluate the trade-off, the transformer depth increased to 3.

B. Transformer Head Count

This hyperparameter defines the number of attention heads per transformer layer. Hrycej, Bermeitinger, and Handschuh demonstrated that multi-head attention exhibits comparable representational power to stacking multiple transformer encoders, particularly in datasets involving animals (e.g., CUB-200-2011, ImageNet) [9]. In accordance with these findings, the head count was increased to 2.

C. Learning Rate

According to Yao. Y, Papakostas. M and Burzo. M et. al's experiment [10], learning rate of 3e-4 in AdamW optimizer has a fine-tuned on each task. Therefore, the learning rate was adjusted from 5e-4 to 3e-4 to achieve more stable training dynamics and to facilitate improved convergence behavior through finer gradient updates.

D. Optimizer

In the original implementation of KITPose, the authors introduced a custom version of the Adam optimizer, incorporating adaptive weight strategies to dynamically modulate gradient contributions based on keypoint visibility [2]. To assess its effectiveness, the optimizer was replaced with the widely adopted AdamW variant, commonly used in transformer training due to its decoupled weight decay formulation.

E. Learning Rate Schedule

Informed by the empirical analysis of Li, Yumer, and Ramanan [11], which highlights the superior performance of cosine and linear scheduling over step scheduling on datasets such as CIFAR-10, a cosine learning rate schedule was employed in place of the original multi-step decay strategy.

F. Scale Factor

As a key parameter influencing the extent of spatial data augmentation and transformation estimation accuracy [12], the scale factor was modified from 0.5 to 0.75 to examine the effect of broader scaling on model generalization and pose estimation accuracy.

DOI: 10.54254/2755-2721/2025.BJ26386

2.2.2. Results and analysis

Table 1. AP-10K (MSD means multi-step decay)

Methods	Backbone	Changed HP	Original	New	AP	AP@0.5	AP@0.75	AP(M)	AP(L)	AR	AR@0.5	AR@0.75	AR(M)	AR(L)
KITPose—E2C4	HRNET-W32	None	None	None	0.701	0.947	0.774	0.575	0.705	0.744	0.955	0.814	0.582	0.747
KITPose—E2C4	HRNET-W32	Depth	2	3	0.961	0.943	0.742	0.485	0.696	0.734	0.949	0.792	0.485	0.738
KITPose—E2C4	HRNET-W32	Head	1	2	0.704	0.942	0.779	0.534	0.707	0.745	0.951	0.819	0.538	0.748
KITPose—E2C4	HRNET-W32	LR-schedule	MSD	Cosin	0.690	0.947	0.756	0.533	0.693	0.734	0.956	0.801	0.542	0.737
KITPose—E2C4	HRNET-W32	LR	5.00E-04	3.00E-04	0.696	0.950	0.761	0.543	0.700	0.737	0.956	0.801	0.546	0.740
KITPose—E2C4	HRNET-W32	Optimizer	Adam	AdamW	0.702	0.943	0.776	0.528	0.706	0.744	0.952	0.819	0.530	0.747
KITPose—E2C4	HRNET-W32	Scale-factor	0.5	0.75	0.696	0.947	0.755	0.583	0.699	0.740	0.954	0.802	0.585	0.743

Table 2. Animalpose (MSD means multi-step decay)

Methods	Backbone	Changed HP	Original	New	AP	AP@0.5	AP@0.75	AP(M)	AP(L)	AR	AR@0.5	AR@0.75	AR(M)	AR(L)
KITPose—E2C5	HRNET-W32	None	None	None	0.697	0.957	0.797	0.657	0.710	0.753	0.961	0.833	0.719	0.764
KITPose—E2C5	HRNET-W32	Depth	2	3	0.684	0.946	0.777	0.656	0.694	0.740	0.952	0.817	0.706	0.751
KITPose—E2C5	HRNET-W32	Head	1	2	0.702	0.957	0.820	0.682	0.709	0.758	0.961	0.851	0.736	0.766
KITPose—E2C5	HRNET-W32	LR-schedule	MSD	Cosin	0.685	0.948	0.794	0.654	0.696	0.742	0.959	0.835	0.721	0.750
KITPose—E2C5	HRNET-W32	LR	5.00E-04	3.00E-04	0.690	0.948	0.769	0.644	0.703	0.749	0.959	0.817	0.710	0.760
KITPose—E2C5	HRNET-W32	Optimizer	Adam	AdamW	0.696	0.948	0.787	0.651	0.710	0.751	0.960	0.829	0.719	0.762
KITPose—E2C5	HRNET-W32	Scale-factor	0.5	0.75	0.697	0.957	0.799	0.660	0.660	0.753	0.961	0.834	0.724	0.763

1) Transformer Depth

Although deeper encoders theoretically enhance representation capacity, the empirical results reveal the practical limitations of over-deepening, aligning with the findings reported by Chen F. and Tsou J. [8]. Increasing depth from 2 to 3 led to consistent declines across both datasets: As reported in Table 1, on AP-10K, AP declined from 0.701 to 0.691, while AP(M) dropped sharply from 0.575 to 0.485. Table 2 shows that a similar pattern emerged on AnimalPose, with AP at 0.684 and AR at

DOI: 10.54254/2755-2721/2025.BJ26386

0.740. These results suggest that the lightweight 2-layer encoder originally adopted in KITPose may strike a better trade-off between expressiveness and overfitting, especially under limited data [2].

2) Transformer Head Count

As the KITPose architecture heavily relies on body-part–guided attention modules [2], the number of attention heads plays a critical role. Increasing head count from 1 to 2 improved AP and AR metrics on both datasets. Table 1 shows that on AP-10K AP@0.75 rose to 0.779 and AR@0.75 reached 0.819—the highest among all variants. Table 2 indicates that on AnimalPose the same configuration achieved the best overall performance, yielding AP = 0.702, AR = 0.758, and AR@0.75 = 0.851. These results follow Hrycej.T etc.'s result[9], indicating that multi-head attention helps localize fine-grained keypoints, especially for large and articulated mammals, which aligns with KITPose's original goal of generalizing across varying body plans.

3) Learning Rate Scheduler

Across both datasets, cosine decay resulted in slight but consistent drops in most metrics. As shown in Table 1, on AP-10K the overall AP@[0.5:0.95] fell from 0.701 to 0.690, and AP(M) dropped markedly by 0.042, while all AR metrics either declined or remained essentially unchanged. Corresponding results in Table 2 reveal that on AnimalPose the cosine decay similarly reduced AP@[0.5:0.95] (from 0.697 to 0.685) and AR (from 0.753 to 0.742), although AR@0.75 and AR(M) experienced a minor increase of 0.002.

These results indicate that cosine scheduling, despite its theoretical smoothness, fails to outperform the original multi-step schedule in the KITPose context[2], which is against Li. M etc.'s result [11]. This may be attributed to the relatively short step-decay training period (e.g., 200-300 epochs), which the "cliff-like" learning rate can quickly enter a more stable convergence interval, while cosine decay is smooth decline, and on small datasets or small models, the training may end before it drops low enough, resulting in insufficient convergence. This enlightens us that the choice of LR scheduler should be combined with the actual dataset, and not the dataset should not be generalized.

Accordingly, the multi-step decay strategy was retained for subsequent optimization experiments.

4) Learning Rate

Overall, reducing the initial learning rate from 5e-4 to 3e-4 did not consistently improve performance across either dataset. As recorded in Table 1, a marginal increase in AP@0.5 on AP-10K was accompanied by slight declines in most other metrics. Table 2 shows that on AnimalPose the deterioration was even more pronounced. These results suggest that a smaller learning rate, which corresponds to smaller update steps, may hinder effective optimization in this context—potentially leading to underfitting or slower convergence to a suboptimal point [13]. Thus, excessively small learning rates may not benefit final performance and should be chosen with caution.

5) Optimizer Type

As reported in Table 2, on AnimalPose the adoption of AdamW produced a noticeable drop (AP@0.5: 0.957 vs. 0.948; AP@0.75: 0.797 vs. 0.787). By contrast, Table 1 shows that on AP-10K performance remained essentially unchanged, the differences being negligible. These results suggest that while AdamW is still competitive, especially in unbalanced annotated datasets, Adam can match or slightly outperform it on more general datasets. Based on this finding, the optimizer remains unchanged to ensure that the entire training process proceeds under the optimal configuration.

6) Scale Factor

The scale factor governs how much training images are randomly zoomed in or out. KITPose originally employed a scale of 0.5 [2]. Increasing it to 0.75 produced mixed results. Table 1 indicates

DOI: 10.54254/2755-2721/2025.BJ26386

that on AP-10K this adjustment lifted AP(M) to 0.583 and AR(M) to 0.585, signifying improved detection of medium-sized objects. Conversely, Table 2 shows that on AnimalPose the same modification produced virtually no impact (AP remained at 0.697), a discrepancy attributable to the dataset's more homogeneous object scales. Thus, while scale-sensitive augmentation helps in high-variance datasets like AP-10K, its benefit may plateau elsewhere.

In the experimental results, the best performance on the AP-10K dataset was achieved with head = 2, yielding a mAP of 0.704 (a slight improvement over the original 0.701). Similarly, on the AnimalPose dataset, the best configuration was also head = 2, with a mAP of 0.701 (an improvement from the original 0.697). These findings indicate a consistent optimal hyperparameter choice across datasets and demonstrate that, while keeping the model structure stable, appropriately adjusting the number of attention heads can lead to steady performance gains.

2.2.3. Summary

The first stage of analysis systematically investigated six common hyperparameters to assess their individual impact on general mammal pose estimation using the KITPose framework. Results revealed that transformer depth, head count, and learning rate exerted the most substantial influence on model performance across both AP-10K and AnimalPose datasets. Specifically, increasing transformer depth beyond two layers led to performance degradation, while adding a second attention head consistently improved keypoint localization, particularly for complex poses. Learning rate adjustments showed dataset-specific sensitivity, indicating potential for further refinement.

In contrast, changes to optimizer type, learning rate scheduler, and scale factor yielded either marginal or inconsistent effects, and thus were deemed less critical for fine-grained tuning. Based on these findings, Stage 2 will focus exclusively on automated hyperparameter optimization using Optuna, targeting only the depth, head count, and learning rate dimensions, while keeping other settings fixed.

2.3. Stage 2: auto optimization

2.3.1. Transition to automated hyperparameter optimization

Manual hyperparameter tuning in Stage 1 provided insights into the impact of different hyperparameters on the KITPose model. However, given that hyperparameter optimization (HPO) can be mathematically formalized as a black-box optimization problem [14], it is challenging to exhaustively explore all possible combinations of hyperparameters and identify the optimal configuration through manual tuning alone [15]. This complexity, especially in higher-dimensional spaces, necessitates the use of automated algorithms to efficiently search the hyperparameter space and ensure reproducibility.

In Stage 2, Optuna—a Bayesian optimization framework—is employed to automate the tuning process. Building on the Stage 1 findings, the search ranges for the following hyperparameters are set:

·TRANSFORMER_DEPTH: From 1 to 4
·TRANSFORMER_HEADS: From 1 to 4
·LR_FACTOR: From 0.05 to 0.5
·MODEL_LR: From 1e-5 to 1e-3

Optuna is utilized to identify the optimal hyperparameter combination within these ranges, thereby further enhancing the performance of the KITPose model.

DOI: 10.54254/2755-2721/2025.BJ26386

2.3.2. Results and analysis

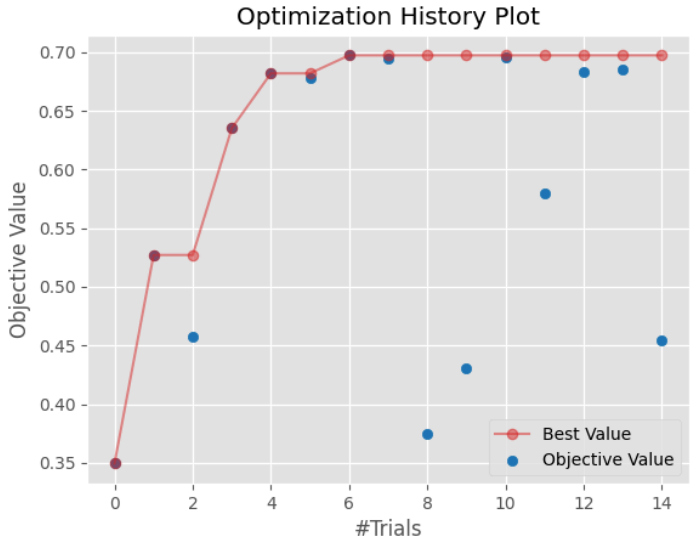

Figure 1. AnimalPose history result

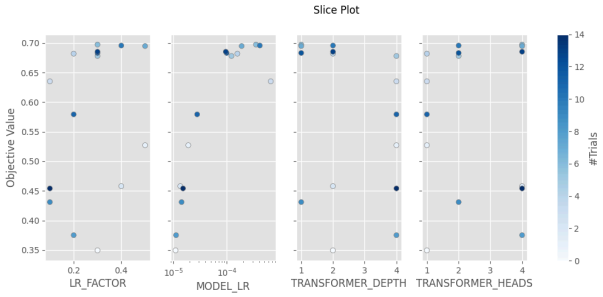

Figure 2. AnimalPose slice result

Figure 1 indicates that on the AnimalPose dataset, KITPose's 15-trial Bayesian hyperparameter search reveals a clear bifurcation in the architectural space. The top-performing configurations are exclusively shallow, multi-head setups (depth $\leq$ 2, heads $\geq$ 4, lr_factor = 0.3–0.4), which consistently push mAP above 0.69 and validation accuracy beyond 0.75. In contrast, any configuration with depth = 4 cluster in the bottom tier, falling over 10 percentage points below the median, irrespective of learning-rate scaling.

Crucially, the performance collapse observed at depth = 4 corroborates the Stage-1 conclusion that increasing encoder depth is detrimental to KITPose training on AnimalPose. This finding aligns with the original KITPose design by Xu. T, Rao. J and Song. X, etc. [2], which adopts a depth of 2, suggesting that a shallow encoder structure is indeed better suited for general mammal pose estimation. However, unlike the top-performing models, the original KITPose architecture employs single-head self-attention, a deliberate choice made to emphasize explicit keypoint-to-body-part interactions over attention diversity or capacity scaling. As described by Xu et al. [2], the design focuses on using full-channel feature slices and straightforward attention pathways to support structure-aware reasoning. Notably, the authors neither empirically compare single-head and multi-head variants nor claim the latter to be inferior, leaving open the possibility that broader attention mechanisms of the proposed model may further enhance performance.

DOI: 10.54254/2755-2721/2025.BJ26386

The experiment result in Figure 2 shows that the relative scales of Model_LR and LR_Factor jointly steer final performance. When both fall in the mid-to-high window (Model_LR ∈ [1e-4, 4e-4], LR_Factor ∈ [0.3, 0.4]), the model reliably reaches about 0.68 mAP. Departing from this window degrade results: a high Model_LR with a low LR_Factor (6.59e-4 vs. 0.1) lowers mAP to 0.635, while an overly small Model_LR paired with an aggressive LR_Factor (1.90e-5 vs. 0.5) pushes it down to 0.527. This finding aligns with Zhang et. Al's result that a smaller learning rate can hinder effective optimization in this case [13]. These observations indicate that the two hyper-parameters interact—rather than operate independently—and must be tuned in concert to maintain stable convergence on AnimalPose.

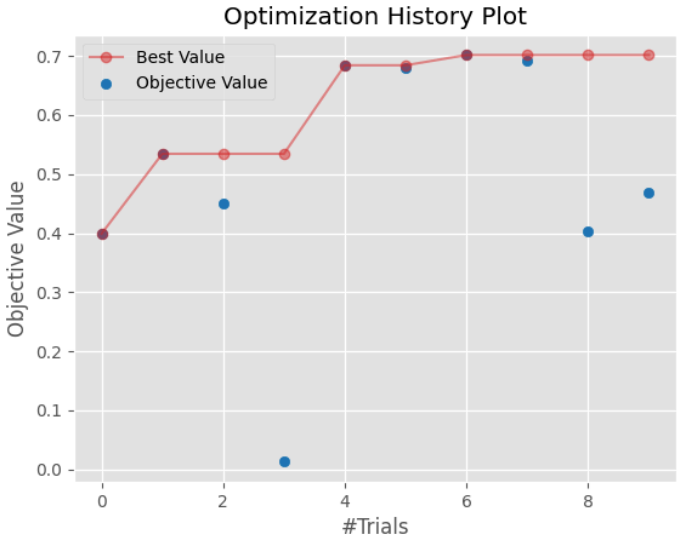

Figure 3. AP-10K history result

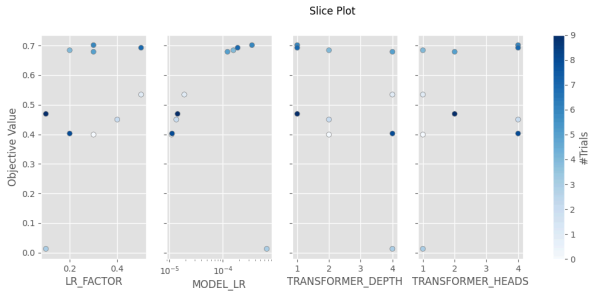

Figure 4. AP-10K slice result

Figure 3 indicates that on the AP-10K dataset, KITPose's 10-trial Bayesian hyper-parameter search again exposes a striking architectural bifurcation that mirrors—yet sharpens—the pattern observed on AnimalPose. The performance frontier is occupied exclusively by configurations with depth ≤ 2 and heads ≥ 2, while any increase in depth to 4 precipitates an immediate collapse. Specifically, the three top-ranked trials (trial 6, 7, 4) share a shallow encoder (depth = 1 or 2) and a 4-head self-attention module, jointly yielding mAP ≥ 0.68 and validation accuracy ≥ 0.79. Conversely, all runs with depth = 4 (trials 1, 3, 5, 8) cluster at the bottom, with the worst performer (trial 3) plummeting to mAP = 0.013—more than 65 percentage points below the median. This catastrophic degradation, independent of learning-rate rescaling, corroborates the Stage-1 finding

DOI: 10.54254/2755-2721/2025.BJ26386

that encoder depth constitutes the dominant failure mode for KITPose on quadruped keypoint benchmarks.

Crucially, the AP-10K results, exhibiting the same trend as those obtained on AnimalPose, reinforce the original KITPose prescription of Xu et al. [2], which indicates that a shallow depth is more suitable for general mammal pose estimation.

Figure 4 further disentangles the coupled influence of Model_LR and LR_Factor. A contiguous high-performance island emerges when Model_LR ∈ [1e-4, 4e-4] and LR_Factor ∈ [0.3, 0.4]—precisely the coordinates of trial 6 (Model_LR = 3.49×10^{-4}, LR_Factor = 0.3) which attains the global optimum (mAP = 0.702), which is outperforms the original experiment with a fixed random seed (22). Departures from this window are systematically penalized: an overly small Model_LR (1.12×10^{-5}) paired with LR_Factor = 0.2 (trial 8) yields mAP = 0.403, and a disproportionately large Model_LR (6.59×10^{-4}) coupled with LR_Factor = 0.1 (trial 3) drives the model into divergence (mAP = 0.013). These observations confirm that Model_LR and LR_Factor interact synergistically; neither can be tuned into isolation without risking convergence instability.

In these experiments, the current best performances are all shown in the same combination of hyperparameters, which depth=1, heads=4, lr_factor=0.3 and model_lr=3.49e-4. In AnimalPose, the result is almost as same as the original result, and in AP-10K, the result is 0.702, which gets a higher performance than the original result.

2.3.3. Summary

In summary, the results are further supported by Hrycej et al. [9], who likewise observed that multi-head attention leads to better performance in pose estimation tasks. Together, these insights underscore that for animal keypoint estimation, architectural breadth—not depth—is the dominant factor in achieving reliable convergence and strong generalization. And the selection of LR should not be extreme, both LR_Factor and Model_LR will have a positive impact on the model when they are in the mid-to-high window.

3. Conclusion

This study systematically investigates the impact of hyperparameter choices on the performance of KITPose, a transformer-based framework for general mammal pose estimation. Through a two-stage experimental design that combines controlled single-variable analysis with Bayesian optimization, transformer depth, attention head count, and learning rate are identified as the most critical levers for model performance. The findings demonstrate that shallow encoders (depth ≤ 2) with multi-head attention (≥ 2 heads) strike an optimal balance between expressiveness and generalization. In contrast, increasing depth to 4 leads to consistent performance collapse across both AP-10K and AnimalPose datasets. Automated tuning further reveals that moderate learning rates (1e-4 to 4e-4) and balanced learning rate factors (0.3 to 0.4) are essential for stable convergence.

These insights not only validate the original architectural decisions in KITPose but also provide a reproducible and extensible protocol for hyperparameter tuning in animal pose estimation. Nevertheless, the exploration was confined to a limited set of parameters and modest trial counts because of computational constraints; more exhaustive searches could reveal further gains. Future work should therefore expand the hyperparameter space and apply similar analyses to other transformer-based pose models.

DOI: 10.54254/2755-2721/2025.BJ26386

References

[1] Rao, J., Xu, T., Song, X., Feng, Z. H., & Wu, X. J. (2022). Kitpose: Keypoint-interactive transformer for animal pose estimation., In Chinese Conference on Pattern Recognition and Computer Vision, Springer, (pp. 660–673).

[2] Xu, T., Rao, J., Song, X., et al. (2025) Learning structure-supporting dependencies via keypoint interactive transformer for general mammal pose estimation. International Journal of Computer Vision, 133: 3858–3876. https: //doi.org/10.1007/s11263-025-02355-0

[3] Cao, J., Tang, H., Fang, H. S., Shen, X., Lu, C., & Tai, Y. W. (2019). Cross-domain adaptation for animal pose estimation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9498-9507).

[4] Sosa, J, Perry, S, Alty, J et al. (2022) Of Mice and Pose: 2D Mouse Pose Estimation from Unlabelled Video Frames using Synthetic Data. In: CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling, 19 - 24 June 2022, New Orleans, United States.

[5] Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., Tao, D. (2021) AP-10K: A benchmark for animal pose estimation in the wild. In: Proceedings of the NeurIPS 2021 Datasets and Benchmarks Track (Round 2). Available at: https: //github.com/AlexTheBad/AP10K

[6] Liao, L., Li, H., Shang, W., Ma, L. (2022) An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. ACM Transactions on Software Engineering and Methodology, 31(3): Article 53, pp. 1–40. https: //doi.org/10.1145/3506695

[7] Bengio, Y. (2012) Practical recommendations for gradient-based training of deep architectures. arXiv preprint, arXiv: 1206.5533. https: //doi.org/10.48550/arXiv.1206.5533.

[8] Chen, F., Tsou, J.Y. (2022) Assessing the effects of convolutional neural network architectural factors on model performance for remote sensing image classification: An in-depth investigation. International Journal of Applied Earth Observation and Geoinformation, 112: 102865. https: //doi.org/10.1016/j.jag.2022.102865

[9] Hrycej, T., Bermeitinger, B., Handschuh, S. (2022) Number of attention heads vs. number of transformer-encoders in computer vision. In: Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) – KDIR. SciTePress, pp. 315–321. https: //doi.org/10.5220/0011578000003335

[10] Yiqun Yao, Michalis Papakostas, Mihai Burzo, Mohamed Abouelenien, and Rada Mihalcea. 2021. MUSER: MUltimodal Stress Detection using Emotion Recognition as an Auxiliary Task In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2714–2725, Online. Association for Computational Linguistics. Available at: https: //aclanthology.org/2021.naacl-main.216

[11] Li. M., Yumer, E., Ramanan, D. (2020) Budgeted training: Rethinking deep neural network training under resource constraints. arXiv preprint, arXiv: 1905.04753. https: //doi.org/10.48550/arXiv.1905.04753

[12] Wu, Y., Zand, M., Etemad, A., & Greenspan, M. (2022, October). Vote from the center: 6 dof pose estimation in rgb-d images by radial keypoint voting. In European Conference on Computer Vision (pp. 335-352). Cham: Springer Nature Switzerland.

[13] K. Zhang, R. Cai, C. Zhou and Y. Liu, et. al (2024) Debiased Contrastive Learning for Time-Series Representation Learning and Fault Detection, In: IEEE Transactions on Industrial Informatics, vol. 20, no. 5, pp. 7641-7653, May 2024, doi: 10.1109/TII.2024.3359409.

[14] Bischi, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., Lindauer, M. (2023) Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. WIREs Data Mining and Knowledge Discovery, 13(2): e1484. https: //doi.org/10.1002/widm.1484

[15] Van Luyen, T., Van Cuong, N. (2024) Metaheuristics-based hyperparameter tuning for convolutional neural networks. In: Thi Dieu Linh, N., Hoang, M.K., Dang, T.H. (Eds.), Ad Hoc Networks. ADHOCNETS 2023. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 558. Springer, Cham. https: //doi.org/10.1007/978-3-031-55993-8_4