Signal, Image and Video Processing (2025) 19:999
https://doi.org/10.1007/s11760-025-04594-7

ORIGINAL PAPER

Exploring Personality in Human-Object Interactions

Yalım Doğan[1] · Sinan Sonlu[1] · Serkan Demirci[1] · Arçin Ülkü Ergüzen[1] · Uğur Güdükbay[1]

Received: 16 June 2025 / Revised: 14 July 2025 / Accepted: 27 July 2025

Abstract
Personality is crucial for accurate and realistic communication in animated humans. Temporal features like body movement help express personality traits. Studies control personality by utilizing high-level motion parameters in general actions, but current research lacks focus on object interaction animations. While object interaction is not a social concept, the subject's personality can affect the motion during interaction. This study examines personality expression in various object interaction sequences to identify the differences due to object types, performed actions, and their iterations. We train a neural motion field-based network to author an animation's intended personality during object interaction, utilizing our personality-aware motion augmentations. We validate our approach with a user study to assess the resulting motions' personality, accuracy, and realism. The results suggest that augmentations better differentiate the positive and negative traits, especially for conscientiousness and extraversion, but at the cost of reduced realism and accuracy. In contrast, data-driven manipulations yield realistic and accurate results, but their impact on personality is subtle. However, when we alter multiple OCEAN factors simultaneously, the resulting changes in the motion are more noticeable.

Keywords 3D Object Interaction · SMPL-X Body Model · OCEAN Personality Model · Neural Motion Reconstruction · Generative Adversarial Networks

1 Introduction

Expressive animations are critical for improved realism and communication in digital media. The expressive features communicate digital actors' personalities, emotions, and intent. Facial expressions, body pose, and gaze contribute to social interactions and emerge as a reaction to the environment. Such reactions exhibit styles based on the subject's personality and emotional behavior. Nonsocial actions such as object interaction are often overlooked in expressive personality literature; however, the stylistic motion during object interaction can also express personality; for instance, a steady gaze and slow movements can indicate high conscientiousness.

This study explores the potential of personality expression using movement style in three-dimensional (3D) human-object interaction sequences. We introduce a novel neural model for personality manipulation of object interaction motions. We use the Five Factor Model for personality, which is also known by the acronym of its five orthogonal factors: OCEAN [1]. Our Neural Motion Field (NeMF)-based generator architecture (Figure 1, left) inputs the object class, action purpose, and target OCEAN factors to transform the input motion to express the desired personality traits. Training of such a generator module is achieved using an adversarial scheme (Figure 1, right) where we utilize a critic module to assess realism of the synthesized motion compared to real ones and a regressor module to ensure the resulting motion consistently represents the intended personality and object semantics. Since expressive human-object interaction datasets lack personality annotations, we introduce them to a customized 3D object interaction dataset. We also use our personality-based motion augmentation framework to increase the sample size and improve the training. Our contributions are as follows:

✉ Uğur Güdükbay
gudukbay@cs.bilkent.edu.tr

Yalım Doğan
yalim.dogan@bilkent.edu.tr

Sinan Sonlu
sinan.sonlu@bilkent.edu.tr

Serkan Demirci
serkan.demirci@bilkent.edu.tr

Arçin Ülkü Ergüzen
ulku.erguzen@bilkent.edu.tr

[1] Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

Published online: 05 September 2025

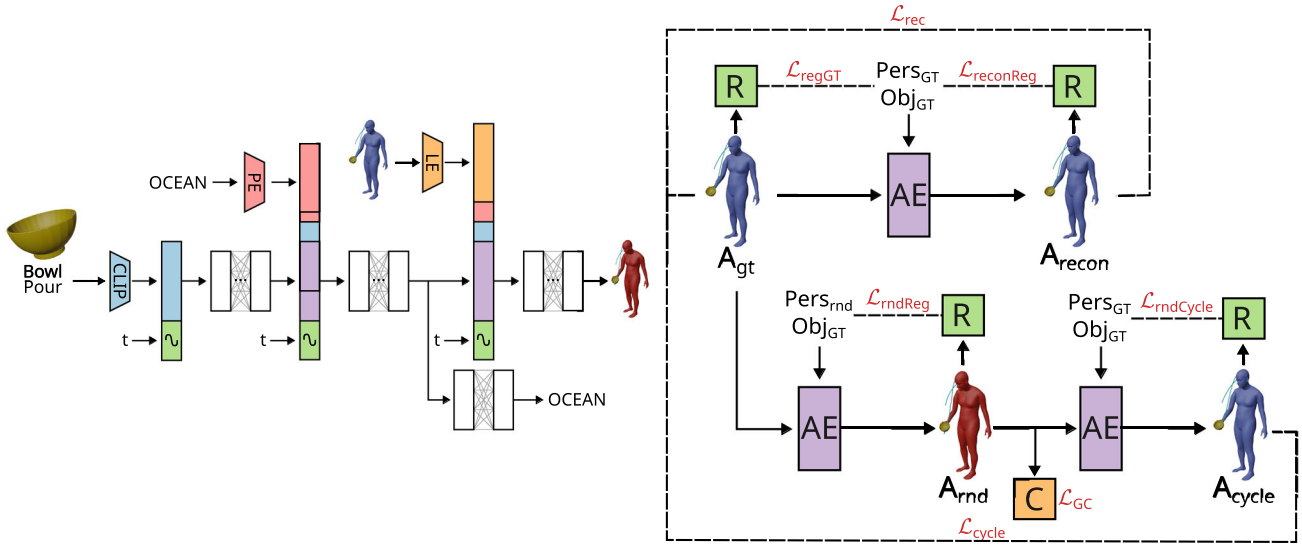

Fig. 1 The proposed neural model for personality manipulation of human-object interaction motions. Left: The neural personality manipulation model. The NeMF-based generator model takes scene features in the following order: object name and purpose, target OCEAN factors, and motion parameters of the source motion. The latent vectors are accompanied by a positional encoded time t. The resulting motion's personality matches the target personality, similar to the output OCEAN factors. Right: During training, *critic* (C) assesses the motion quality while *regressor* (R) assesses its semantic accuracy in multiple stages. Both *critic* and *regressor* are trained together with the generator (AE)

- We examine object interaction datasets to introduce our object and body-aware motion augmentation, which utilizes Laban Movement Analysis (LMA) to introduce controlled variance to object interaction motion to increase the training sample size.
- We use the OCEAN personality model to label a set of object interaction animations using crowdsourcing, where we use the collected information in training.
- We introduce a neural motion-based generative adversarial network to author the personalities in provided motions. The network also contains modules for inferring the existing personality, object type, and action for provided motions.
- We conduct additional user studies to compare the realism and expressiveness of augmentation and network-based motions. We alter each personality factor separately to isolate its effects on the motion style.

2 Related Work

Computers store animation as sequences of keyframes, which, when interpolated, appear as moving objects. In the case of human animation, each keyframe represents a body pose as a configuration of articulated 3D joints. These joint configurations drive skinned 3D meshes that appear as representations of the human body. SMPL-X (A Skinned Multi-Person Linear Model - eXpressive) [2] introduces a parametric human body utilizing data-driven morphing that can represent arbitrary body types. SMPL-X representation also includes finger joints, which can accurately represent object grasping.

Motion capture helps produce human animation by recording real-life actors' movements. The actors wear special suits with per-joint motion sensors or are captured using a multi-camera setup with optional depth sensors to synthesize high-quality data. This process is expensive, and even using state-of-the-art tools, capturing the movements of the objects that actors interact with is challenging due to overlaps.

Especially when handling small objects, a significant portion of the object's surface becomes hidden, complicating optical recognition. Consequently, the available object interaction datasets are limited to specific objects and involve highly restricted actions [3]. Certain sets utilize egocentric recording to capture better the interacted object, which does not allow the reproduction of the actor's full pose.

Additionally, many object interaction datasets focus on detection and do not include 3D data, which is essential for our purpose. 3D reconstruction of such video-only datasets usually yields low-quality results due to the arbitrary nature of the actions; the reconstruction results in shaky and noisy motions, especially when objects are thrown into the mix. We use GRAB [4] in this work due to its high-quality 3D data with sufficient repetitions of the same action by multiple actors with diverse body types.

Motion authoring can be achieved using precise controls or style transfer by providing an additional motion that contains the target style expected to be realistically incorporated

into the original motion. This process requires extracting style information using an encoder architecture and conditioning the synthesized motion accordingly [5]. However, this approach requires style motions that precisely reflect the desired attributes, which may not be compatible with the original motion's semantic context. In this case, precise control via high-level traits provides a feasible interface for motion authoring.

LMA-based high-level control of apparent personality traits is a well-studied approach for authoring motion's psychological content [6, 7]. LMA Effort parameters can be interpreted as various motion edits that alter the movement's style. While many previous works utilize handcrafted adjustments for LMA-based personality changes, data-driven solutions are scarce as a comprehensive LMA motion dataset is lacking. To overcome this issue, we use LMA-based augmentation to introduce personality variation to GRAB.

3 Dataset

We opted for the GRAB dataset, which includes non-articulated object interactions, to minimize reconstruction errors. Personality expression is not the primary goal of GRAB.

The subjects do not aim to convey personality traits or emotions; such expressive features, if any, occur naturally. Since object interaction happens in a nonsocial context, detecting an expressive style in arbitrary samples is challenging. Consequently, we manually filtered the GRAB dataset to 10 basic household items with identifiable contexts. We also removed two hand interaction samples for simplicity, leaving us with 10 object categories and 102 sequences, which we give further detail about in Appendix A.

To reduce gender bias, we used a gender-neutral SMPL-X model to visualize the animations. We also remove facial expressions and use the default beta parameters for each animation. However, this alters body proportions, such as height and arm length. This change does not cause problems in isolated motions, but extra steps are required to preserve the object's motion. To this end, we first fix the subject's fingers to their initial articulations and stick the object to the corresponding hand based on its original orientation relative to the wrist. Sticking is achieved via utilizing Blender's [8] internal constraint logic. As the initial hand location would be altered due to changed body proportions, the distance between "neutralized" and the original hand is used as an offset for the object. This approach ensures the subject handles the object in a consistent form and constant contact throughout the entire sequence.

3.1 Motion Augmentation

Due to the low sample size for each action, we developed a motion augmentation framework using Blender. The framework alters input motion regarding joint rotations and temporal resolution, following adjustments based on previous work [9]. For authoring, we used LMA Effort parameters: *Space*, *Time*, *Flow*, and *Weight*, with each parameter being in the $[-1, 1]$ range. Further details on each LMA Effort augmentation are available in Appendix B.

Space controls the distance between hand and foot joints and is declared as factor f_s. Positive f_s increases the distance between each hand and feet along the frontal axis, while negative f_s brings them closer together. This behavior results in "indirect," thus, "spreading" motion for positive and "direct," thus, "enclosing" for negative values.

Weight modifies vertical posture while keeping feet grounded. Increasing f_w imposes "heavy" weight on the subject, causing descent, while decreasing it causes the subject to stand taller with "light" weight. This parameter is essential for conveying the weight of the interacted object.

Time changes the playback speed of the animation, depending on the delta changes between consecutive frames of hands and feet. The lower the delta changes, the slower the movement, and the more aggressive the speed adjustment.

Flow reduces the number of keyframes without changing the animation duration if the provided factor value f_f is negative. We achieve this using Blender's internal *Decimate Keyframes* operation. Further reducing f_f causes the animation to become more "bounded" as more keyframes are reduced to interpolated animation curves.

The original motion sequences from GRAB contain the object transformations independent of the subject's joint rotations. Any LMA Effort augmentation also requires the object to be adjusted accordingly. The aforementioned contact-fixing process solves this problem; it ensures the object follows the hand position during the interaction. We crop the sequences such that the object is always in contact with the subject's hand to maintain physical interaction realism.

3.2 Annotation Framework

We developed a 3D tool using Unity Engine to annotate the motion sequences. For better interpretability, we added a visualization for the resulting trajectory of the object as a post-processing step. The participants were shown five sliders representing each trait of the OCEAN personality model. Each trait can be annotated in scale $[-3, 3]$, which is the normalized form of the Ten-Item Personality Inventory (TIPI) [10]. The participants could rotate around and zoom into the object and control the time of the animation (see Figure 2).

We asked additional questions to rate the motion realism, action label accuracy, object attributes (temperature,

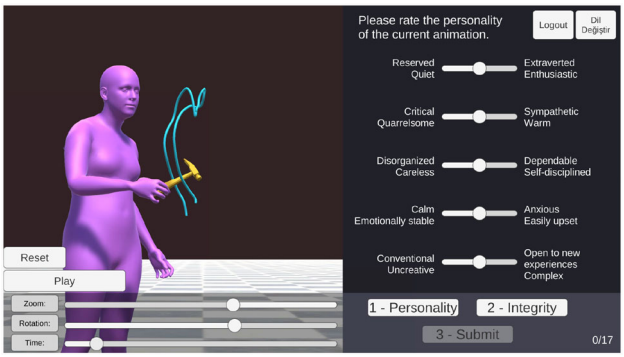

Fig. 2 Our motion annotation tool. The blue lines represent the trajectory of the object's center

weight, softness), and the animation elements that influence the participant's decision the most (body pose, gaze, object trajectory, or finger articulations).

4 Neural Motion Manipulation

We utilized a Variational Auto Encoder (VAE) neural network based on NeMF by He et al. [11] to manipulate a provided object interaction motion sequence via altering its apparent personality. The original architecture encodes every motion into two latent vectors, local and global components. The local component is responsible for representing the joint rotations, positions, and velocities, while the global component contains the orientation of the body and its translation. For our purposes, the subject does not alter its translation during the motion in a noticeable fashion; therefore, we discarded the global component and maintained the global features from the original motion. In the original model, the network accepted a constant number of frames, 128, which we respected in this work.

Our NeMF-based generator architecture consists of two parts: encoders for each aspect of the motion and a fully connected decoder module to emit different features of the synthesized motions. For encoders, we split features of the motion as: *Local Encoder* for local features, *Object Encoder* for object information, and *Personality Encoder* for target personality. All encoders' output features are independently forwarded into separate linear layers for distribution calculation to abide by the variational properties of the architecture via the reparameterization trick.

Local encoder takes features for each joint, namely position and velocity, including angular and global rotations, according to the subject's pelvis. Position and velocities are represented in 3D vectors, while global rotations are represented in 6D rotations [12]. NeMF's local encoder was based on the SMPL model [13], where finger joints are not considered. We excluded them when using the SMPL-X [2] model, as the subject's fingers are locked, as mentioned in previous sections. The joint features are concatenated in the time dimension t and processed using a predefined graph-based skeleton convolution [14].

Our *object encoder* takes the object transformation in the following form: object intent, and name. We do not use the object mesh and object transformation, as we have fixed finger articulations. As object intent and name are initially in text format, we preprocessed them using CLIP [15] into fixed-size (512) latent vectors, as in [16]. After all features are concatenated into a single vector in the time domain, they are processed using multiple residual blocks, similar to the global encoder in the original NeMF architecture. Each residual block contains Conv-BatchNorm-Activation layers, accompanied by residual connections. As NeMF expects features to be provided across all motion frames, we repeat the obtained latent representations for intent and object name for the expected number of frames.

Personality encoder takes OCEAN personality factors as input. Each OCEAN factor with the original scale of $[-3, 3]$ is normalized to $[-1, 1]$ before proceeding. OCEAN is constant across the frames, like object intent; hence, it is repeated across frames. The personality encoder has a similar architecture to the object encoder; it only varies in the number of residual blocks and input dimensionality.

The base NeMF decoder has multiple residual blocks, where each encoder is introduced to the network as input at different steps. The latent vector to be fed to each layer is initialized with positional encoding to provide a temporal relation between each frame for the decoder. The first encoder's latent, "local" in the original design, is concatenated to the current latent vector. For an empirically decided number of upcoming layers, this latent vector is concatenated with the immediate output of each layer using skip connections. After a particular encoder's layers are processed, the latest feature vector is fed to a fully connected network as output. The next encoder's latent is concatenated to the current latent vector, and the process is repeated for the next set of layers. Here, the order of the encoders implies a dependency, where local latent variables came earlier than global in the original NeMF. In our work, we determined order as *Object*, *Personality*, and lastly *Local*.

During the training process, the generator first produces the original motion by taking its original personality as input. We apply a reconstruction loss, $\mathcal{L}_{rec}$, to ensure the generated motion matches the original. Next, to ensure the model's capability of transferring between arbitrary personalities for the same motion while maintaining its core characteristics, we introduce cycle loss $\mathcal{L}_{cycle}$ to the generator. The generator generates a random motion from a random personality, which is then transferred back to the original motion via its original personality. The "cycled back" motion contributes to the generator's overall loss via reconstruction losses. For

both $\mathcal{L}_{rec}$ and $\mathcal{L}_{cycle}$, we use the following reconstruction terms:

Position loss ($\mathcal{L}_{pos}$) : $L2$ norm of the difference between predicted and ground-truth joint positions. λ_{pos} is set as 20 during training.

Rotation loss ($\mathcal{L}_{rot}$) : Geodesic distance between predicted and ground-truth 6D joint rotations. λ_{rot} is set as 7 during training.

Orientation loss ($\mathcal{L}_{ori}$) : Geodesic distance between predicted and ground-truth root orientation. λ_{ori} is set as 2 during training.

Joint velocity loss ($\mathcal{L}_{vel}$) : $L2$ norm of the difference between predicted and ground-truth joint velocities. λ_{vel} is set as 1 during training.

Angular velocity loss ($\mathcal{L}_{avel}$) : $L2$ norm of the difference between predicted and ground-truth angular velocities. λ_{avel} is set as 1 during training.

KL divergence loss ($\mathcal{L}_{KL}$) : Measures the difference between the learned latent distribution and the prior distribution. λ_{KL} is set as 10^{-4} during training.

OCEAN loss ($\mathcal{L}_{OCEAN}$) : $L2$ norm of the difference between the predicted and ground truth OCEAN personality factors. λ_{OCEAN} is set as 2 during training.

$$\begin{aligned}\mathcal{L}_{rec/cycle} = &\lambda_{rot}\mathcal{L}_{rot} + \lambda_{pos}\mathcal{L}_{pos} + \lambda_{ori}\mathcal{L}_{ori} \\ &+ \lambda_{vel}\mathcal{L}_{vel} + \lambda_{avel}\mathcal{L}_{avel} + \lambda_{KL}\mathcal{L}_{KL} \\ &+ \lambda_{OCEAN}\mathcal{L}_{OCEAN}\end{aligned} \tag{1}$$

To ensure the local motion is influenced by the provided target personality, we employ an additional adversarial methodology on top of our generator architecture. We utilize Wasserstein GAN [17] with a gradient penalty (GP) approach to ensure the generator can synthesize motion consistent with the target personality even when fed with random, unseen personalities. To this end, we train a separate critic network within our training loop, which takes a real and generated motion to distinguish their distributions. In this sense, the generator tries to synthesize motions with random personalities that are indistinguishable from real ones. However, the critic tries to maximize the difference between distributions calculated as Earth Mover's distance, as in adversarial training. Real motions receive a much higher score than synthesized motions.

Gradually updated updates for the critic are also subject to their losses to ensure stable training. As part of the gradient penalty, synthesized motions are randomly interpolated with real motions and fed to the critic. The calculated gradients' norm is penalized via its associated loss function, which enforces Lipschitz continuity. This way, the critic is deterred from destabilizing the training process due to large fluctuations in gradient updates. The critic also provides feedback on generated motions by providing negative loss to the generator, where the generator is expected to increase the synthetic motion's score. The critic implementation is similar to the generator's; it consists of identical encoders and fewer fully connected blocks.

$$\mathcal{L}_C = \frac{1}{n}\sum_{i=1}^{n} C(\hat{\mathbf{x}}_i) - \frac{1}{n}\sum_{i=1}^{n} C(\mathbf{x}_i) + \lambda_{gp}\mathcal{L}_{gp} \tag{2}$$

$$\mathcal{L}_{gp} = \frac{1}{n}\sum_{i=1}^{n} \left(\left\| \nabla_{\tilde{\mathbf{x}}_i} C(\tilde{\mathbf{x}}_i) \right\|_2 - 1\right)^2 \tag{3}$$

$$\mathcal{L}_{GC} = -\frac{1}{n}\sum_{i=1}^{n} C(\hat{\mathbf{x}}_i) \tag{4}$$

where C is the critic, $\mathbf{x}_i$ denotes the true sample, $\hat{\mathbf{x}}_i$ denotes generated (fake) sample, and $\tilde{\mathbf{x}}_i$ is an interpolated sample used for gradient penalty. λ_{gp} was set to 10 during training.

In addition to the critic, we trained a separate module solely to provide regression feedback for apparent personality, object, and action types. This regressor module works based on local features of the motion and is utilized in multiple stages. Before the generator's losses are calculated, the regressor is fed real motions and their ground truth personality, object type, and action labels. The regressed values are used as the regressor's losses. Then, each motion synthesized by the aforementioned generator labels is regressed by the regressor and compared against ground truth values. Motions synthesized with ground truth, random labels, and cycled motions are subject to separate loss terms:

OCEAN Regression loss ($\mathcal{L}_{OCEANReg}$): $L2$ norm of the difference between the predicted and ground truth OCEAN personality factors. $\lambda_{OCEANReg}$ is set as 10 during training.

Object name loss ($\mathcal{L}_{name}$): Cross-entropy loss between the predicted and ground truth object name. λ_{name} is set as 5 during training.

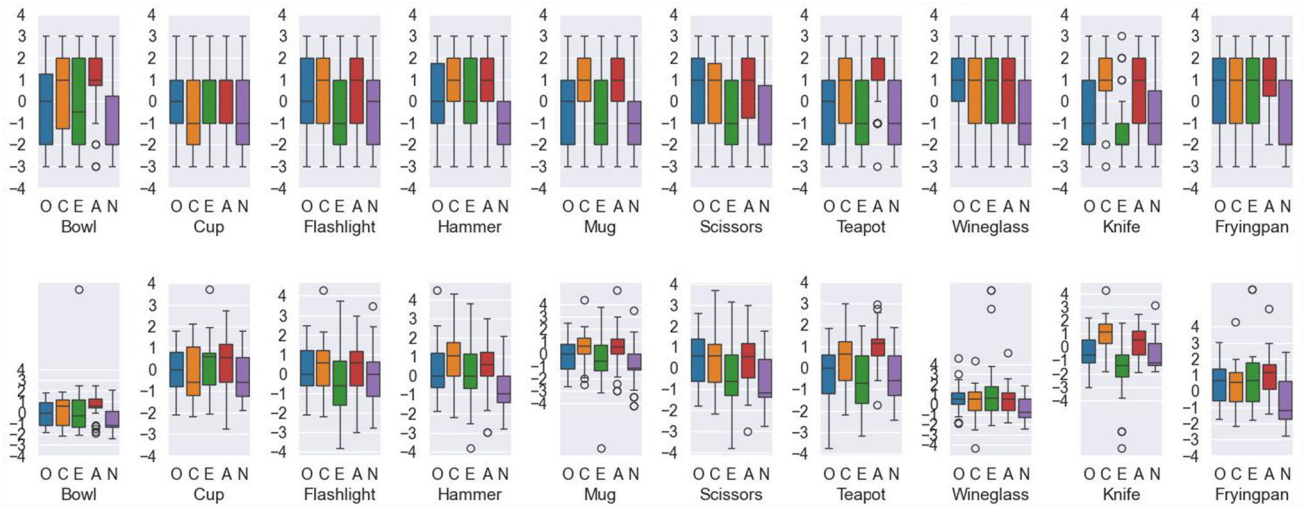

Fig. 3 Raw (top) and standardized (bottom) annotations for each object in the first study. The variance for each OCEAN factor decreases after standardization, thus certain annotations become outliers while maintaining the overall diversity

Object intent loss ($\mathcal{L}_{intent}$): Cross-entropy loss between the predicted and ground truth object intent. λ_{intent} is set as 5 during training.

$$\mathcal{L}_{reg} = \lambda_{OCEANReg}\mathcal{L}_{OCEANReg} + \lambda_{name}\mathcal{L}_{name} + \lambda_{intent}\mathcal{L}_{intent} \tag{5}$$

$$\mathcal{L}_{GR} = \lambda_{cycleReg}\mathcal{L}_{cycleReg} + \lambda_{reconReg}\mathcal{L}_{reconReg} + \lambda_{randomReg}\mathcal{L}_{randomReg} \tag{6}$$

$$\mathcal{L}_{G} = \mathcal{L}_{rec} + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{critic}\mathcal{L}_{GC} + \mathcal{L}_{GR} \tag{7}$$

where $\mathcal{L}_{reg}$ is applied to the regressor while $\mathcal{L}_{GR}$ is added to the overall loss of the generator ($\mathcal{L}_G$). $\mathcal{L}_{cycleReg}$, $\mathcal{L}_{reconReg}$, and $\mathcal{L}_{randomReg}$ all share the same formulation with $\mathcal{L}_{reg}$, where they only differ in input motion. We set $\lambda_{cycleReg}$, $\lambda_{reconReg}$, and $\lambda_{randomReg}$ as 5, λ_{cycle} as 0.1, and λ_{critic} as 1 for training.

We applied Xavier initialization for training and used Adamax optimizer for all modules. For *Generator* and *Regressor*, we used learning rate of 10^{-4} and 2×10^{-5} for *Critic* with one iteration per generator iteration. We trained all modules for 200 epochs with a batch size of 32 using an A100 Nvidia GPU. We only applied weight decay with 10^{-7} to the generator module.

5 Experiments and Discussion

We performed two user studies: one for dataset annotation and training, and another for assessing the quality of the generated animations. The details of user studies, including participant demographics, can be found in Appendix D.

5.1 Dataset Annotation and Training

Fifty online participants rated the personality of the chosen samples in the first study, resulting in an average of eight annotations per sample. Each participant annotated 17 randomly selected samples in an average of 96 seconds per sample. We examine the personality distribution of the different object categories in Figure 3, with additional results on realism and integrity available in Appendix E. The top row depicts the raw data, and the bottom row shows the results after standardization using the learning approach for fitting a normal distribution with $std = 1$ for each participant's answers and object category per *OCEAN* factor.

While specific object categories like *Hammer* and *Mug* deviate from the neutral personality, especially for conscientiousness and agreeableness, we observe less variance for openness and extraversion. This behavior is expected as object interaction is less social and has less of an intellectual focus, which these traits mainly represent. In contrast, the perceived effect of conscientiousness highly relates to the attention towards the object of interest, as expected.

We use standardized annotations to train our generator. To increase the scale of our limited dataset, we also applied augmentation across all LMA Efforts independently with values {-0.75, 0, 0.75}. Upon augmentation, the resulting OCEAN is also subject to change due to changes in motion. For this purpose, we used the well-known mapping between OCEAN

input controls and the LMA Effort augmentations in PERFORM [6]. To achieve LMA Effort to OCEAN mapping, we transposed the normalization axis and kept the significant portions of the matrix. As each augmentation alters a single LMA Effort, we applied delta modifications to the original OCEAN. This approach dramatically increases our motion count by 80 while introducing subtle diversity.

As our network accepts motions with a fixed length of 128 frames, we applied a sliding-window approach to each motion. We repeat the last frame if the motion is shorter than the fixed length. After training our model, we apply latent optimization to novel motions during synthesis. The first step of the optimization extracts the latent vectors for all inputs. Then, while fixing personality and object latent variables, the local latent is optimized to maximize reconstruction performance according to the original motion. Then, to ensure proper personality alteration, the process is repeated to minimize the loss from *regressor* module according to the target personality. In our experiments, we optimized the reconstruction target personality for 500 and 100 steps, respectively. We depict the effect of agreeableness and openness on the resulting motion in Figure 4, and Appendix C illustrates the remaining factors.

5.2 Motion Control User Study

During our second user study, we selected a single motion to represent each object by finding the one with the set of OCEAN factors closest to 0. Then, participants compared five versions of each motion with identical semantics, only varying in their personalities. One motion is kept as original, while the others are altered versions. Two of the motions' random, single traits are set to -1 and 1 using the augmentation module, while the other two are set via the network. Eighty-three online participants rated the generated samples in our second user study, corresponding to an average of 16 samples for each comparison. Details can be found in Appendix F.

We report Tukey's Honestly Significant Difference (HSD) adjusted pairwise mean differences for the significant effects in Table 1. We compare the mean scores of the models in contrast to the base model and among each other. We expect the successful combinations to deviate significantly from the base samples and their opposite counterparts. We only include the personality measurements that the network aims to alter. Although we also observe a difference in the factors other than the one the network focuses on, we leave analyzing such correlations as future work. We exclude cases where ANOVA does not indicate a significant effect; unabridged results are available in Appendix G.

We generally observe that augmentations yield more apparent differences. The factor that is most influenced by the adjustment is conscientiousness, followed by extraversion and neuroticism. We observe that the effect on perceived

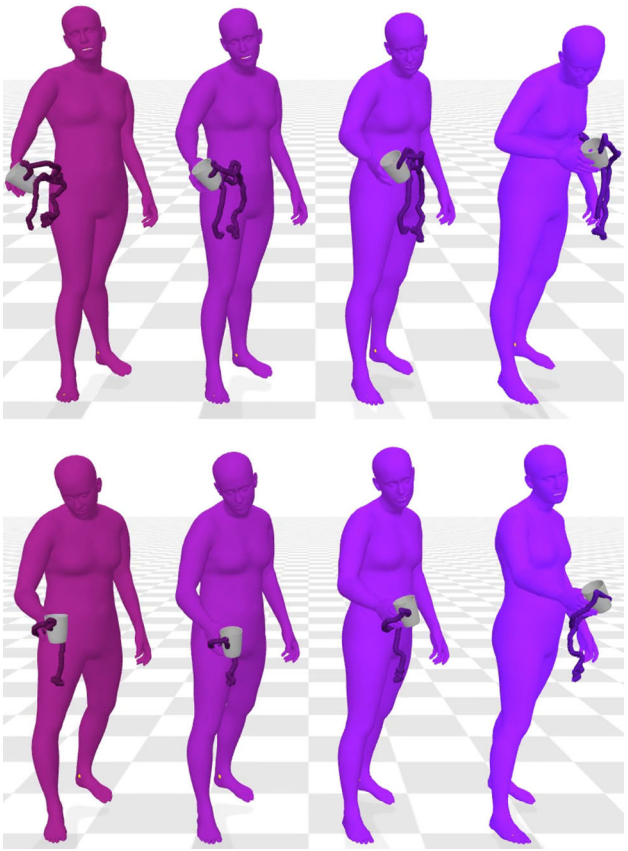

Fig. 4 Personality effect on the network output for agreeableness (top) and openness (bottom). The factor ranges from -1 (left) to 1 (right) to alter movements and object trajectory, while other factors are constant

agreeableness is minimal. Object categories such as bowl and scissors produce more expressive results, likely because these motions give more freedom to the actor. In contrast, categories like *Hammer* have connotations that do not leave much room for style to emerge. For example, the first user study revealed that animations in the *Hammer* category were highly conscientious. Thus, manipulation to alter this effect had less opportunity than an initially more neutral category.

We observe that augmentations create a difference between negative and positive samples by exaggerating the negative case for conscientiousness and the positive case for extraversion. However, this creates a decrease in realism. In particular, the augmentations for positive traits significantly decrease both realism and action accuracy. In contrast, the network's output, especially for the positive manipulation, has significantly higher realism and accuracy. For the negative manipulation, both approaches have similar results regarding realism and accuracy; however, augmentations yield more inconsistent results between positive and negative cases.

We observe that the network's effect is limited in altering the personality significantly; however, this results in less

Table 1 Tukey HSD adjusted p-values (ρ) and the mean differences (Δ) between the different models for each object category and OCEAN factor, excluding insignificant outcomes. Each column compares different model pairs: **Base** for original animation, **ANeg** and **APos** represent negative and positive changes using the augmentation framework, respectively. **NNeg** and **NPos** represent the same changes applied using the motion authoring network, respectively. We examine the factor that the system aims to alter for each personality group; for example, when the system aims to change openness, we evaluate the performance based on perceived openness. Realism (Re.) and Accuracy (Ac.) scores are calculated across all objects. Significant results are colored with Green for $\rho < 0.05$, and Blue for $\rho < 0.1$. We excluded Agreeableness values as we found no significant results associated with them

	Category	Base–ANeg		Base–NNeg		Base–APos		Base–NPos		ANeg–NNeg		ANeg–APos		NNeg–NPos		APos–NPos	
		ρ	Δ	ρ	Δ	ρ	Δ	ρ	Δ	ρ	Δ	ρ	Δ	ρ	Δ	ρ	Δ
O	*Teapot*	.510	1.286	.074	2.143	.019	2.571	.510	1.286	.827	.857	.510	1.286	.827	-.857	.510	-1.286
C	*Bowl*	.000	-3.062	.627	-.938	.686	-.875	.916	-.562	.018	2.125	.013	2.188	.980	.375	.990	.312
	Cup	.551	-1.067	.729	.867	.911	.600	.671	.933	.055	1.933	.133	1.667	.999	.067	.989	.333
	Hammer	.054	-2.214	.388	-1.429	.992	-.357	.819	-.857	.861	.786	.149	1.857	.952	.571	.970	-.500
	Knife	.010	-2.067	.548	-.933	.999	.133	.940	.467	.351	1.133	.005	2.200	.160	1.400	.982	.333
	Mug	.060	-2.214	.624	-1.143	.868	-.786	.999	-.143	.679	1.071	.404	1.429	.732	1.000	.932	.643
	Scissors	.010	-2.533	.395	-1.333	.984	-.400	.900	-.667	.502	1.200	.045	2.133	.900	.667	.997	-.267
	Teapot	.633	-1.200	.885	-.800	.885	-.800	.913	-.733	.990	.400	.990	.400	.999	.067	.999	.067
	Wineglass	.048	-2.214	.165	-1.786	.809	-.857	.137	-1.857	.982	.429	.422	1.357	.999	-.071	.707	-1.000
	Drinking	.000	-3.062	.627	-.938	.686	-.875	.916	-.562	.018	2.125	.013	2.188	.980	.375	.990	.312
	Pouring	.551	-1.067	.729	.867	.911	.600	.671	.933	.055	1.933	.133	1.667	.999	.067	.989	.333
E	*Bowl*	.999	-.067	.999	-.133	.088	1.933	.779	.867	.999	-.067	.072	2.000	.675	1.000	.620	-1.067
	Cup	.905	-.667	.905	.667	.094	1.933	.933	.600	.410	1.333	.009	2.600	.999	-.067	.410	-1.333
	Flashlight	.848	-.750	.999	.000	.305	1.438	.986	.375	.848	.750	.033	2.188	.986	.375	.607	-1.062
	Knife	.989	-.333	.999	.000	.006	2.467	.962	.467	.989	.333	.001	2.800	.962	.467	.041	-2.000
	Teapot	.995	.333	.999	.000	.082	2.200	.982	.467	.995	-.333	.191	1.867	.982	.467	.256	-1.733
	Drinking	.999	-.067	.999	-.133	.088	1.933	.779	.867	.999	-.067	.072	2.000	.675	1.000	.620	-1.067
	Pouring	.905	-.667	.905	.667	.094	1.933	.933	.600	.410	1.333	.009	2.600	.999	-.067	.410	-1.333
N	*Bowl*	.999	.000	.838	.733	.000	3.667	.791	.800	.838	.733	.000	3.667	.999	.067	.001	-2.867
	Scissors	.999	-.125	.976	.438	.040	2.125	.708	.938	.940	.562	.025	2.250	.960	.500	.494	-1.188
	Drinking	.999	.000	.838	.733	.000	3.667	.791	.800	.838	.733	.000	3.667	.999	.067	.001	-2.867
	Pouring	.999	-.200	.999	-.200	.868	.800	.900	-.733	.999	.000	.745	1.000	.967	-.533	.349	-1.533
Re.	*All*	.000	-.549	.000	-.768	.000	-1.217	.000	-.705	.387	-.219	.000	-.668	.986	.063	.000	.513
Ac.	*All*	.144	-.049	.016	-.066	.000	-.136	.271	-.042	.928	-.017	.000	-.087	.796	.024	.000	.094

disruption in motion realism. The data-driven nature of the network helps the synthesized animation to appear more realistic and accurate. The network can alter the personality most successfully for conscientiousness. We further explore the effects of authoring personality using multiple OCEAN factors in Appendix C.

5.3 Ablation and Comparison Studies

To evaluate the contribution of each component in our network, we performed ablation studies on quantitative metrics in Table 2. For reconstruction performance, we calculated *Joint Rotation Error* **(JRE)** and *Joint Position Error* **(JPE)**. JRE is calculated for all body joints, except the hands, as geodesic loss and is reported in degrees. JPE is calculated as the mean square distance between joints and reported in centimeters. For assessing the motion synthesis performance, we utilized *Diversity* [18] and *Fréchet Inception Distance* **(FID)** [19]. The diversity metric is calculated as the average of the latent distances between different motion instances. A high diversity value indicates highly distinct motions. FID is calculated as the Fréchet distance between latents of synthetic and real motions, thus representing the similarity between their distributions. A low FID score indicates that the synthetic motions are similar to real motions. To calculate both metrics, we used a pre-trained NeMF network. We fine-tuned the NeMF using the GRAB [4] dataset following NeMF's original training protocol. As our network only uses local features, we discarded the global motion component of NeMF. As for personality, we calculate the *Mean Square Error* (MSE) difference between the intended personality and the input. The intended personality is calculated using a regressor that is trained together with all components enabled. We also calculate the accuracy of the output motion in terms of inferred object names and action types.

We turned off a distinct module at each network variation, keeping others intact. In addition to disabling modules, we also reversed the order of the scene features provided to the NeMF module to observe the effect of order dependency between latents. The results in Table 2 indicate that the cycle consistency is adversarial to our network in terms of both motion reconstruction and personality recognition. However, a higher FID value than the *All* configuration indicates a diverted distribution compared to real motions, although object interaction metrics are solid. Every other network configuration, while generating improvement in some areas, fails at generating semantically consistent interaction sequences.

To our knowledge, no personality authoring networks involving object interaction sequences exist in the literature. Therefore, we chose the closest architecture regarding object interaction generation: IMoS by Ghosh et al. [16]. The IMoS

Table 2 Ablation results for our NeMF-based network. Except for the "w/o cycle" option, all network versions suffer from semantic consistency regarding object interaction. The last two columns show the accuracy of the output motion in terms of inferred object names ("Name Acc.") and action types ("Action Acc."). The row "reverse latent order (rev. lat. ord.)" shows the results when the order of the scene features provided to the NeMF module is reversed to observe the effect of order dependency between latents

Models	Motion Rec.		Motion Syn.		Personality Recognition					Object Interaction	
	JRE° ↓	JPE (cm) ↓	Div ↑	FID ↓	O ↓	C ↓	E ↓	A ↓	N ↓	*All*	Name Acc. ↑	Action Acc. ↑
w/o critic	7.55	0.315	**4.95**	12.24	**0.255**	0.388	0.437	0.393	0.396	0.3738	27.0	30.0
w/o cycle	**1.44**	**0.01**	4.03	4.93	0.369	**0.195**	**0.178**	0.351	**0.362**	**0.291**	99.0	**100.0**
w/o regressor	28.82	1.305	4.11	25.45	0.467	1.048	0.216	0.206	1.079	0.6	29.0	37.0
rev. lat. ord.	7.99	0.19	4.47	10.86	0.313	0.370	0.201	**0.196**	1.938	0.6	28.0	35.0
All	1.69	0.016	4.19	**2.94**	0.340	0.205	0.475	0.279	0.454	0.35	**100.0**	**100.0**

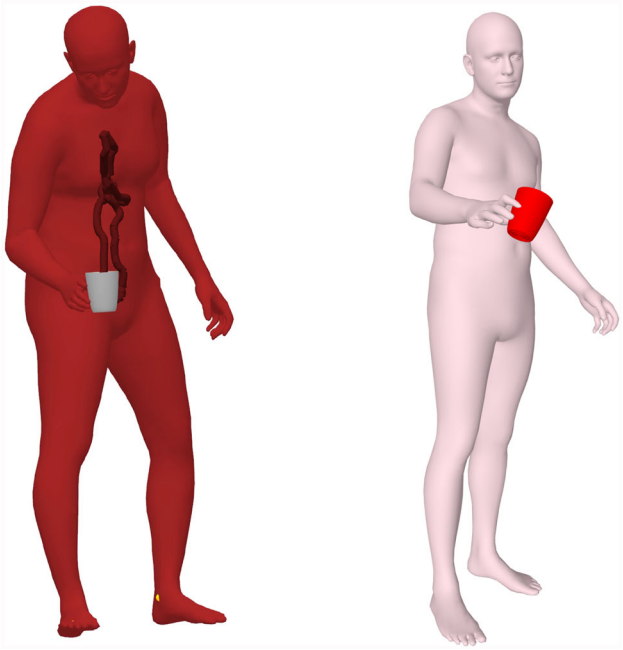

Fig. 5 The IMoS generates motions that lack personality and take considerable time, considering their focus on object handling. Left: our network, right: IMoS result

framework generates consecutive object interaction frames over a history buffer for a given object. Even though the object name and the action type are inputs similar to ours, their method does not allow personality authoring. Additionally, their method takes considerable time due to object optimization. Our method takes 45 seconds to alter the personality, and IMoS takes 3-4 minutes to generate a motion on Nvidia 3070Ti, including all optimization protocols. Due to high time consumption, they also require interpolation between the limited number of generated frames. Although our method snaps the object to fingers and generates a fixed number of frames, it allows personality authoring, thus increasing the animation's expressiveness via high-level control. As seen in Figure 5, the animation generated by IMoS lacks expressive elements.

6 Conclusion

This study focuses on manipulating personality perception in human animations with object interaction. Unlike general style or personality transfer in animation, object interaction introduces more constraints over the process. The object's motion regarding the figure's contact with the object requires careful inspection. We introduce a personality-based augmentation pipeline to control the personality-based animation style while keeping the object's motion integrity. We

then train a neural manipulation system to apply the changes automatically.

We evaluate both approaches through a perception study and report their performances per object category and personality factor. While applying the manipulation through the augmentation system better distinguishes the positive and negative traits, it also disturbs the perceived realism and motion accuracy. In contrast, the network produces much better realism and accuracy at the cost of more subtle personality variation. However, in case multiple OCEAN factors are altered simultaneously, such differences become easily noticeable.

Future work could examine our additional annotations for object attributes about the perceived personality. Increasing the diversity and scale of annotated interaction sequences, detailed finger articulations, and body movement could improve training performance and, consequently, the expressive capabilities of our network. The animation element each participant focuses on could impact their personality estimations, which could reveal interesting insights.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11760-025-04594-7.

Author Contributions Y.D. designed and implemented the proposed framework and performed the experimental study. S.S. and S.D. contributed to the implementation and experimental study. A.Ü.E. contributed to implementing motion augmentation and the experimental study. U.G. supervised the study. All authors contributed to the writing and reviewing of the manuscript.

Funding This research is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 122E123.

Data Availability No datasets were generated or analysed during the current study.

Code availability Our object interaction dataset annotations, LMA-based augmentation framework, and personality-based motion manipulation network are publicly available in https://github.com/YalimD/ObjectInteractionPersonalityAuthoring.

Declarations

Conflicts of Interest The authors declare no conflict of interest.

Ethical Approval Bilkent University Ethical Committee for Human Research approved the study with the decision number İAEK_2024_07_26_01.

References

1. McCrae, R.R., Costa, P.T.: Personality in Adulthood: A Five-factor Theory Perspective. Guilford Press, New York, NY, USA (2005)
2. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR '19, pp. 10975–10985. IEEE, Piscataway, NJ, USA (2019)
3. Antoun, M., Asmar, D.: Human object interaction detection: Design and survey. Image Vis. Comput. **130**, 15 (2023)
4. Taheri, O., Ghorbani, N., Black, M.J., Tzionas, D.: GRAB: A dataset of whole-body human grasping of objects. In: European Conference on Computer Vision - ECCV 2020. Lecture Notes in Computer Science, vol. 12349, pp. 581–600. Springer, Cham, Switzerland (2020)
5. Li, P., Aberman, K., Zhang, Z., Hanocka, R., Sorkine-Hornung, O.: GANimator: Neural motion synthesis from a single sequence. ACM Transactions on Graphics **41**(4), 2 (2022)
6. Durupinar, F., Kapadia, M., Deutsch, S., Neff, M., Badler, N.I.: PERFORM: Perceptual approach for adding OCEAN personality to human motion using Laban Movement Analysis. ACM Transactions on Graphics **36**(1), 16 (2016)
7. Sonlu, S., Güdükbay, U., Durupinar, F.: A conversational agent framework with multi-modal personality expression. ACM Transactions on Graphics **40**(1), 16 (2021)
8. Blender Foundation: Blender - a 3D Modelling and Rendering Package. (2018). http://www.blender.org
9. Ergüzen, A.Ü., Demirci, S., Sonlu, S., Gudukbay, U.: Personality Transfer in Human Animation: Handcrafted Versus Data-Driven Approaches. Available at SSRN: https://dx.doi.org/10.2139/ssrn.5003652 (2025)
10. Gosling, S.D., Rentfrow, P.J., Swann, W.B., Jr.: A very brief measure of the big-five personality domains. J. Res. Pers. **37**(6), 504–528 (2003)
11. He, C., Saito, J., Zachary, J., Rushmeier, H., Zhou, Y.: NeMF: Neural motion fields for kinematic animation. In: Proceedings of the 36th Conference on Neural Information Processing Systems. NeurIPS '22, pp. 4244–4256 (2022)
12. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5745–5753. IEEE, Piscataway, NJ, USA (2019)
13. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Transactions on Graphics **34**(6), 16 (2015)
14. Fussell, L., Bergamin, K., Holden, D.: Supertrack: Motion tracking for physically simulated characters using supervised learning. ACM Transactions on Graphics **40**(6), 13 (2021)
15. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., : Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning. ICML '21, pp. 8748–8763 (2021). PMLR
16. Ghosh, A., Dabral, R., Golyanik, V., Theobalt, C., Slusallek, P.: IMoS: Intent-driven full-body motion synthesis for human-object interactions. Computer Graphics Forum **42**(2) (2023). Wiley Online Library. Article no. cgf.14739, 12 pages
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. Adv. Neural. Inf. Process. Syst. **30**, 5769–5779 (2017)
18. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. MM '20, pp. 2021–2029. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394171.3413635
19. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a Nash equilibrium. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, pp. 6629–6640. Curran Associates Inc., Red Hook, NY, USA (2017)