OrthoFuse: Training-free Riemannian Fusion of Orthogonal Style-Concept Adapters for Diffusion Models

Supplementary Material

A. Smoothness

In this section, we prove that the set of $\mathcal{GS}$ orthogonal matrices form a smooth manifold using a slightly different notation which is more convenient for these purposes. To do this, let us define additional objects we are going to use further.

For $i = 1, \dots, m$ let B_i be a block-diagonal matrix with k_i orthogonal blocks of size $b_i \times b_i$ and let P_i for $i = 1, \dots, m-1$ be an arbitrary fixed permutation matrix. Based on these notation, we define a set of orthogonal matrices $\mathcal{A}_m^{\text{orth}}$:

$$\mathcal{A}_m^{\text{orth}} = \{A \mid A = B_m P_{m-1} \dots B_1\}.$$

These set of matrices is a significantly more general set than $\mathcal{GS}$ orthogonal matrices, considered in Section 3.2 due to the arbitrariness of m, the choice of every permutation matrix and the choice of the number of blocks in each B_i. For this set we provide our main technical contribution.

Theorem 3. *$\mathcal{A}_2^{orth}$ is a submanifold in $\mathrm{O}(N)$ and in $\mathrm{GL}_N(\mathbb{R})$.*

Proof. To prove this, let us firstly define groups $\mathcal{B}_i$:

$$\mathcal{B}_i = \mathrm{O}(n_1) \times \dots \times \mathrm{O}(n_{k_i}) \tag{14}$$

and their Cartesian product $G = \mathcal{B}_2 \times \mathcal{B}_1$. Note that G is a Lie group as it is a Cartesian product of two Lie groups. Additionally, G is compact because it is closed (being defined by the system of closed polynomial equations $AA^\top = I$) and bounded (each element of the Cartesian product is bounded by the square root of the matrix size in the Frobenius norm).

Note that the orthogonality condition implies that the transpose operation preserves the block-diagonal structure and orthogonality. It also gives us an opportunity to define Lie group action: $G = \mathcal{B}_2 \times \mathcal{B}_1 \curvearrowright M$ where M is a manifold. In our case we can choose $M = \mathrm{O}(N)$ or $M = \mathrm{GL}_N(\mathbb{R})$ and take $x = P_1 \in M$. The rule of action is defined as follows:

$$g \cdot x : \mathcal{B}_2 \times \mathcal{B}_1 \times M \to M : B_2 x B_1^\top. \tag{15}$$

Let us prove that it is really action by definition:

- $g \cdot x \in M$ because orthogonal matrices are closed under multiplication.
- $(g'g) \cdot x = B_2' B_2 x (B_1' B_1)^\top = B_2' B_2 x B_1^\top (B_1')^\top = g' \cdot (g \cdot x)$.
- $e \cdot x = I_2 \cdot x \cdot I_1^\top = x \cdot I_1^\top = x$.
- The action is smooth as the composition of polynomial (hence smooth) matrix operations.

Thus, by definition ([1, 16]), this is indeed a Lie group action.

Using the fact that the action of a compact Lie group on a manifold (either $\mathrm{GL}_N(\mathbb{R})$ or $\mathrm{O}(N)$) yields submanifold (see [1, Theorem 2.3] or [16, Corollary 21.6 & Problem 21-17]), we obtain the desired result. In addition to that, using the relation concerning group orbits we get: $\mathrm{Orb}_G(x) \cong G/\mathrm{Stab}_G(x)$, where

$$\begin{aligned}\mathrm{Stab}_G(P_1) = \{(B_2, B_1) \in \mathcal{B}_2 \times \mathcal{B}_1 \mid \\ B_2 P_1 B_1^\top = P_1 \iff B_2 = P_1 B_1 P_1^\top\}\end{aligned} \tag{16}$$

and $\cong$ denotes diffeomorphism (see the proof in, e.g., [16]). □

Remark 1. *We note that Theorem 3 is more general than our particular use case: diagonal blocks of B_i can be of different size and belong to* U *(unitary matrices),* SO *(special orthogonal matrices), or* SU *(special unitary matrices). Moreover, the theorem statement is true for any matrix P_1 taken from the manifold M (not necessarily a permutation matrix).*

Since we have shown that $\mathcal{A}_2^{\text{orth}}$ is a submanifold, multiplying on the left (or right) by a fixed permutation matrix is a smooth diffeomorphism, and therefore the image $\{ P_L A P_R \mid A \in \mathcal{A}_2^{\text{orth}} \}$ is also a submanifold for any fixed permutation matrices P_L, P_R.

B. $\mathcal{GS}$-orthogonal matrices merging

The main problem in $\mathcal{GS}$-orthogonal matrices merging is that the orbit of the action is diffeomorphic to a homogeneous space. It means that the same matrix in manifold M can be obtained in several ways. Here Perfect Shuffle permutation plays an important role: on the one hand, it provides good mixing, and on the other hand, its similarity transformation allows us to rather easily compute the stabilizer, i.e., obtain a description of the orbit. For the latter, we recall the result from [4, 8].

Lemma 1 ([4, 8])**.** *Let P be a Perfect Shuffle permutation matrix. For any diagonal matrix D of size $k_1 \times k_1$ and any matrix M of size $b_1 \times b_1$, the following equation holds:*

$$P(D \otimes M)P^\top = M \otimes D.$$

From Lemma 1, it follows that when $k_1 \geq b_2$ the stabilizer is discrete. Consequently, by [16, Theorem 21.17, Theorem 21.18], the discrete stabilizer implies that the orbit attains its maximal possible dimension:

$$\dim \mathrm{Orb}_G(P_1) = \dim \frac{G}{\mathrm{Stab}_G(P_1)} = \\ = \dim G - \dim \mathrm{Stab}_G(P_1) \leq \dim G - 0 = \dim G. \tag{17}$$

The following lemma formally clarifies the desired result.

Lemma 2. *Stabilizer is discrete when $k_1 \geq b_2$.*

Proof. To prove that stabilizer is discrete, we need to solve

$$B_2 = P_1 B_1 P_1^\top, \tag{18}$$

where

$$B_2 = \mathrm{diag}(B_2^1, B_2^2, \ldots, B_2^{k_2}), \quad B_2^i \in \mathrm{O}(b_2) \tag{19}$$

Let us denote $C = P_1 B_1 P_1^\top$. In terms of index notation, block-diagonal property for B_2 means that

$$(B_2)_{ij} \neq 0 \implies \left\lfloor \frac{i-1}{b_2} \right\rfloor = \left\lfloor \frac{j-1}{b_2} \right\rfloor$$

Now consider an element $(B_2^l)_{i'j'}$ within a block of B_2, with local indices $i', j' \in \{1, \ldots, b_2\}$ and global indices $i = (l-1)b_2 + i'$, $j = (l-1)b_2 + j'$. According to Lemma 1, to be non-zero, (i,j)-th of B_2 must satisfy $i \equiv j \pmod{k_1}$, and thus $i' \equiv j' \pmod{k_1}$. In the case $k_1 \geq b_2$ the condition $i' \equiv j' \pmod{k_1}$ for $i', j' \in \{1, \ldots, b_2\}$ holds only when $i' = j'$ since $|i' - j'| < b_2 \leq k_1$. Thus, the matrix B_2^l can have nonzero elements only on the diagonal. Since B_2^l is orthogonal, it must be a diagonal orthogonal matrix. The diagonal elements of such a matrix are ± 1. This set is finite and, as any stabilizer, forms a subgroup of G [16]. The same result can be obtained for the case of blocks from SO. □

For the cases where $k_1 \geq b_2$ does not hold, the stabilizer (in general) is non-discrete.

Remark 2. *Note that the Perfect Shuffle permutation is not the only matrix yielding maximum dimension under the condition $k_1 \geq b_2$. However, Perfect Shuffle is optimal from the dense matrix formation perspective, as it minimizes the number of nonzero elements in the resulting matrix [9]. For these practical reasons, we employ Perfect Shuffle in our structured representation.*

Now, knowing the structure of the manifold, we can find the geodesics. To do this, we need some additional theory from [7, 16].

Definition B.1 ([7, 16])**.** *Let M and $\tilde{M}$ be two manifolds. A map $p : \tilde{M} \to M$ is a smooth covering map if:*

- *p is smooth and surjective,*
- *for every point $x \in M$, there exists a neighborhood U of x in M such that $p^{-1}(U)$ is a disjoint union $\bigcup_{i \in I} U_i$ of open subsets of $\tilde{M}$, and for each $i \in I$, the restriction $p : U_i \to U$ is a diffeomorphism.*

Definition B.2 ([7])**.** *Let (M, g) and $(\tilde{M}, h)$ be two Riemannian manifolds. A map $p : \tilde{M} \to M$ is a Riemannian covering map if:*

- *p is a smooth covering map,*
- *p is a local isometry.*

While the formal definitions of isometry and local isometry are available in [7], they will not be essential for our subsequent development.

Proposition 4 ([16], Proposition 21.28)**.** *Every discrete subgroup of a Lie group is a closed Lie subgroup of dimension zero.*

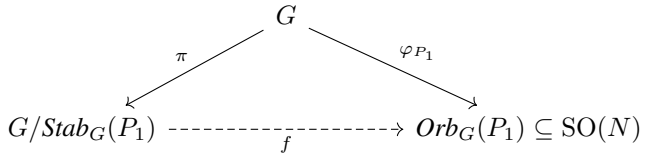
Proposition 5 ([16], Proposition 21.34)**.** *For each $n \geq 1$, the Lie groups $\mathrm{SO}(n), \mathrm{U}(n), \mathrm{SU}(n)$ are connected.*

Proposition 6 ([16], Theorem 21.29)**.** *If G is a connected Lie group and $\Gamma \subseteq G$ is a discrete subgroup, then G/Γ is a smooth manifold and the quotient map $\pi : G \to G/\Gamma$ is a smooth normal covering map.*

Proposition 7 ([7], Proposition 2.18)**.** *Let $p : N \to M$ be a smooth covering map. For any Riemannian metric g on M, there exists a unique Riemannian metric h on N such that p is a Riemannian covering map.*

Proposition 8 ([7], Proposition 2.81)**.** *Let $p : (N, h) \to (M, g)$ be a Riemannian covering map. The geodesics of (M, g) are the projections of the geodesics of (N, h), and the geodesics of (N, h) are the liftings of those of (M, g).*

Proposition 9 ([1, 16])**.** *The following diagram is commutative:*



where $\pi : G \to G/H$ is surjective map that sends each element $g \in G$ to its corresponding coset gH (H denotes $Stab_G(P_1)$): $\pi(g) = gH$. $\varphi_{P_1} : G \to Orb_G(P_1)$ is the map defined by action $\varphi_{P_1}(g) = g \cdot P_1 = B_2 P_1 B_1^\top$. Finally, $f : G/H \to Orb_G(P_1)$ is the diffeomorphism defined by the rule: $f(gH) := g \cdot P_1$.

Proof. Commutativity of the diagram means that for every $g \in G$, the following identity holds:

$$\varphi_{P_1}(g) = f(\pi(g)).$$

Verification: By the definition of π: $\pi(g) = gH$. By the definition of f: $f(\pi(g)) = f(gH) = g \cdot P_1$. By the definition of φ_{P_1}: $\varphi_{P_1}(g) = g \cdot P_1$.

Thus, $\varphi_{P_1}(g) = f(\pi(g))$ for all $g \in G$, which proves the commutativity of the diagram. □

Now let us combine these statements for the case $k_1 \geq b_2$. In this case, the stabilizer forms a subgroup.

Therefore, using Propositions 4, 5, the fact that the Cartesian product of connected Lie groups is connected, and Proposition 6, we conclude that $\pi : G \to G/H$ is a smooth covering map, where $H = \mathrm{Stab}_G(P_1)$.

Next, we show that the action map φ_{P_1} is also a smooth covering map. To do this, it is sufficient to show, by Proposition 9, that the composition of the diffeomorphism f and the smooth covering map π is a smooth covering map. Since π is a smooth covering, for any $\bar{x} \in G/H$ there exists an evenly covered open neighborhood $U \subset G/H$ such that $\pi^{-1}(U) = \bigsqcup_{i \in I} V_i$, where each $V_i \subset G$ is open and the restriction $\pi|_{V_i} : V_i \to U$ is a diffeomorphism. Because f is a diffeomorphism, $W = f(U)$ is an open neighborhood of $f(\bar{x})$ in $\mathrm{Orb}_G(P_1)$. Then:

$$\varphi_{P_1}^{-1}(W) = \pi^{-1}(f^{-1}(W)) = \pi^{-1}(U) = \bigsqcup_{i \in I} V_i.$$

For each i, the restriction $\varphi_{P_1}|_{V_i} = f \circ \pi|_{V_i}$ is a diffeomorphism $V_i \to W$, as it is a composition of two diffeomorphisms. Thus, φ_{P_1} satisfies the exact definition of a smooth covering map.

Finally, let g be the standard Riemannian metric on $\mathrm{SO}(N)$ induced by the Frobenius inner product. As $\mathrm{Orb}_G(P_1)$ is an embedded submanifold, it inherits a Riemannian metric g_{orb}. According to Proposition 7, there exists a unique Riemannian metric h on G such that $\varphi_{P_1} : (G, h) \to (\mathrm{Orb}_G(P_1), g_{\text{orb}})$ is a Riemannian covering map. By Proposition 8, exact geodesics on the orbit are projections of exact geodesics in (G, h). Such metric need not necessarily be identical across blocks, nor necessarily coincide with the Frobenius norm. Nevertheless, we can perform a block-wise connection (uniform across blocks) using formula (4) and empirically verify that the resulting curve exhibits nearly constant velocity in the Frobenius norm, consistent with the constant-speed property of geodesics (see, [7, Definition 2.77]), thereby supporting its interpretation as a meaningful geodesic approximation with respect to the Frobenius norm. These findings are visually confirmed in Figure 5.

Note that another natural approach is to follow a geodesic in the ambient space $\mathrm{SO}(N)$; however, this is computationally expensive, inefficient, and generally does not stay within the $\mathcal{GS}$ manifold. Nevertheless, we empirically demonstrate that the resulting block-wise curve closely approximates a geodesic in the ambient space $\mathrm{SO}(N)$ (see Figure 6).

Observe that also that to obtain the group action we added transposition on the right, but we are given matrices in the form $L_1 P R_1 = L_1 P (R_1^\top)^\top$ and $L_2 P R_2 = L_2 P (R_2^\top)^\top$, and that's why we need to make additional transposition after connecting $R_1^\top$ and $R_2^\top$. Below we prove that this approach is equivalent to directly combining the matrix blocks without introducing additional transposition.

Proposition 10. *For $A_1, A_2 \in \mathrm{SO}(n)$ let*

$$\gamma_{A_1, A_2}(t) = A_1 \exp\left(t \log(A_1^\top A_2)\right), \qquad t \in [0, 1],$$

be the standard geodesic joining A_1 to A_2. Construct the analogous geodesic between the transposed matrices,

$$\gamma_{A_1^\top, A_2^\top}(t) = A_1^\top \exp\left(t \log(A_1 A_2^\top)\right),$$

and transpose the result. Then

$$\gamma_{A_1, A_2}(t) = \left[\gamma_{A_1^\top, A_2^\top}(t)\right]^\top.$$

Proof.

$$\left[\gamma_{A_1^\top, A_2^\top}(t)\right]^\top = \exp\left(t \log(A_1 A_2^\top)\right)^\top A_1.$$

Now let's use $(\exp M)^\top = \exp(M^\top)$ and $(\log M)^\top = \log(M^\top)$ (this follows from the absolute convergence of the Taylor series):

$$\left[\gamma_{A_1^\top, A_2^\top}(t)\right]^\top = \exp\left(t \log(A_2 A_1^\top)\right) A_1.$$

Let $\Omega = \log(A_1^\top A_2) \in \mathrm{SO}(n)$. Since

$$\begin{aligned} A_1 \exp(\Omega) A_1^\top &= A_1 \left(\sum_{k=0}^{\infty} \frac{\Omega^k}{k!} \right) A_1^\top \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} A_1 \Omega^k A_1^\top = \sum_{k=0}^{\infty} \frac{1}{k!} (A_1 \Omega A_1^\top)^k \\ &= \exp(A_1 \Omega A_1^\top). \end{aligned}$$

A similar matrix equation is true for the logarithm function: for $A_1 \in \mathrm{SO}(N)$ we have

$$A_1 \log(\Omega) A_1 = \log(A_1 \Omega A_1).$$

Then

$$A_2 A_1^\top = A_1 \exp(\Omega) A_1^\top = \exp(A_1 \Omega A_1^\top).$$

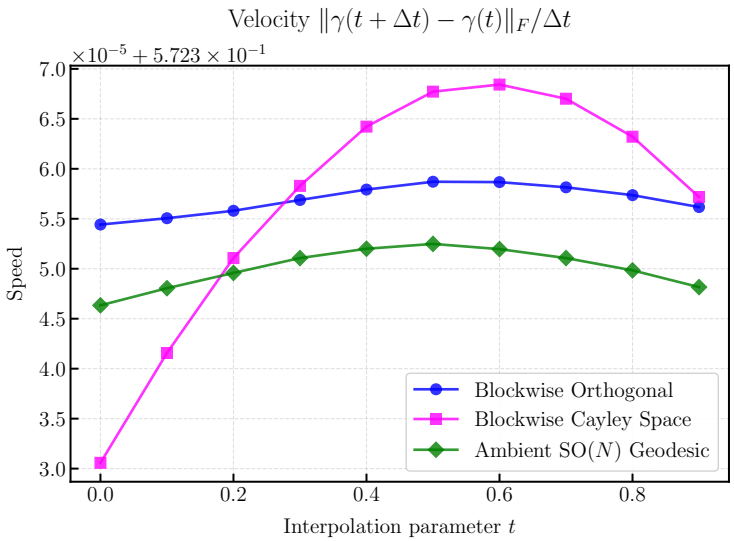


Figure 5. Velocity of geodesics measured in the Frobenius norm: comparison of the exact SO(N) geodesic, the block-wise orthogonal approximation, and the block-wise Cayley space approximation.

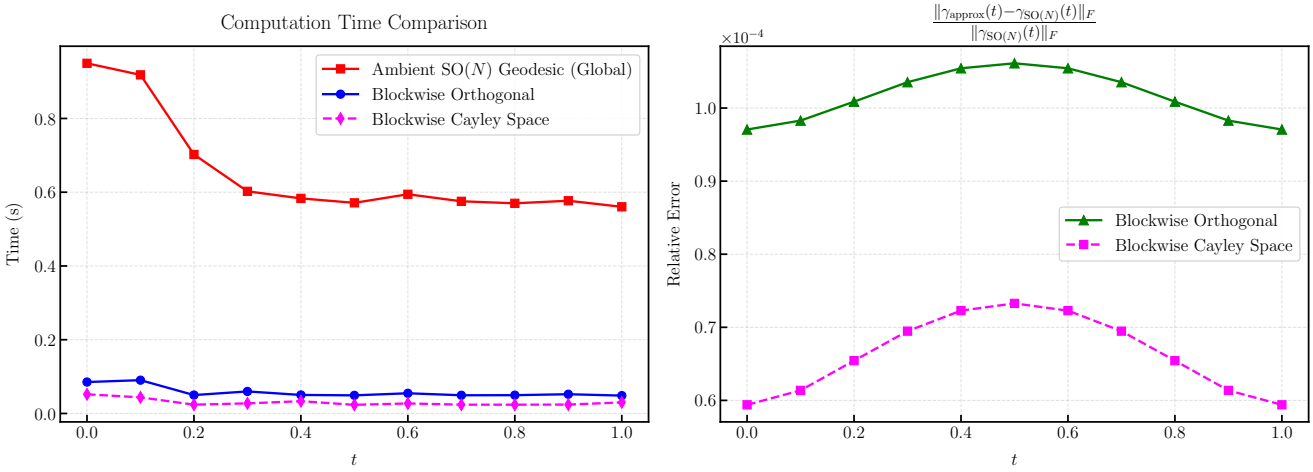


Figure 6. Relative error (in Frobenius norm) between the blockwise geodesic (geodesic projection from G) and the geodesic in M, as well as the computation time of the two methods.

Thus

$$\log(A_2A_1^\top) = A_1\Omega A_1^\top = A_1\log(A_1^\top A_2)A_1^\top.$$

Finally, we obtain:

$$\begin{aligned}\gamma_{A_1^\top,A_2^\top}(t) =& \exp\big(t\log(A_2A_1^\top)\big)\,A_1 = \\ =& \exp\left(tA_1\log(A_1^\top A_2)A_1^\top\right)A_1 = \\ =& A_1\exp\left(t\log(A_1^\top A_2)\right)A_1^\top A_1 = \\ =& A_1\exp\big(t\log(A_1^\top A_2)\big) = \gamma_{A_1,A_2}(t).\end{aligned} \tag{20}$$

□

C. Proof of Proposition 1

Proof. Utilizing eigendecomposition of $B(t)$

$$B(t) = U\Lambda U^*, \quad \Lambda = \operatorname{diag}(x_1+iy_1,\dots,x_n+iy_n), \tag{21}$$

satisfying $|x_i|^2+|y_i|^2=1$. Since the eigenvalues of $B(t)$ lie on a unit sphere, we can express them as $e^{i\phi_i}$ for each $i=1,\dots,n$. Then, for the principle branch of the logarithm, we have

$$\begin{aligned}\log(B(t)) =& U\log(\Lambda)U^* = \\ =& U\log(\operatorname{diag}(e^{i\phi_1},\dots e^{i\phi_n}))U^* = \\ =& U\operatorname{diag}(i\phi_1,\dots i\phi_n)U^*.\end{aligned} \tag{22}$$

Now let us consider $\frac{B(t)-B(t)^\top}{2}$. Using eigendecomposition, we have

$$\frac{B(t)-B(t)^\top}{2} = U\,\mathrm{diag}(iy_1,\dots iy_n)U^*. \tag{23}$$

Subtracting from (22) the final form of (23), we obtain

$$\begin{aligned}\log(B(t)) - \frac{B(t)-B(t)^\top}{2} = \\ = U\,\mathrm{diag}\left(i(\phi_1-y_1),\dots i(\phi_n-y_n)\right)U^*\end{aligned} \tag{24}$$

Using that $y_i = \sin\phi_i$, and assuming that y_i is close to 0 (which is satisfied in practice), we can utilize Taylor expansion for sin function and finally obtain

$$\begin{aligned}&U\,\mathrm{diag}\left(i(\phi_1-y_1),\dots,i(\phi_n-y_n)\right)U^* = \\ &= U\,\mathrm{diag}\left(\mathcal{O}(\phi_1^2),\dots,\mathcal{O}(\phi_n^2)\right)U^* = \\ &= U\,\mathrm{diag}\left(\mathcal{O}(y_1^2),\dots,\mathcal{O}(y_n^2)\right)U^* = \\ &= \mathcal{O}\left(\|\,\mathrm{diag}(y_1,\dots,y_n)\|_2^2\right),\end{aligned} \tag{25}$$

where the latter matrix belongs to $\mathcal{O}\left(\|B(t)-I\|_2^2\right)$. Indeed,

$$\begin{aligned}&\|B(t)-I\|_2^2 = \\ &= \|\,\mathrm{diag}(x_1-1+iy_1,\dots,x_n-1+iy_n)\|_2^2 = \\ &= \|\,\mathrm{diag}(\sqrt{1-y_1^2}-1+iy_1,\dots, \\ &\quad \sqrt{1-y_n^2}-1+iy_n)\|_2^2 \geqslant \\ &\geqslant 1/2\|\,\mathrm{diag}(y_1,\dots,y_n)\|_2^2\end{aligned} \tag{26}$$

for small enough $\max_i |y_i|$, which completes the proof. □

D. Proof of Proposition 3

Proof. Set skew-symmetric matrix $K(t) := \frac{B(t)-B(t)^\top}{2}$. From proposition 1 we have:

$$\log B(t) = K(t) + \mathcal{O}\left(\|B(t)-I\|_2^2\right).$$

Hence,

$$\begin{aligned}B_{\text{Rotated}}(t) = \exp\left(\eta(t)\log B(t)\right) = \\ = \exp\left(\eta(t)K(t) + \mathcal{O}(\|B-I\|_2^2)\right).\end{aligned} \tag{27}$$

In last equation we used that $\eta(t)$ has a closed form $1 + 4t(1-t)$, which is bounded on [0, 1]. Using smoothness of the matrix exponential we get:

$$B_{\text{Rotated}}(t) = \exp(\eta(t)K(t)) + \mathcal{O}\left(\|B(t)-I\|_2^2\right).$$

From Proposition 2, with $\eta(t)$ we obtain the following:

$$\begin{aligned}\exp(\eta(t)K(t)) = \left(I - \frac{\eta(t)}{2}K(t)\right)^{-1}\left(I + \frac{\eta(t)}{2}K(t)\right) + \\ + \mathcal{O}\left(\|\eta(t)K(t)\|_2^2\right).\end{aligned} \tag{28}$$

As $\|\eta(t)K(t)\|_2 = \mathcal{O}(\|K(t)\|_2) = \mathcal{O}(\|B(t)-I\|_2)$, we obtain

$$\exp(\eta(t)K(t)) = B_{\text{OrthoFuse}}(t) + \mathcal{O}\left(\|B(t)-I\|_2^2\right).$$

Combining all the results, we get

$$B_{\text{OrthoFuse}}(t) = B_{\text{Rotated}}(t) + \mathcal{O}\left(\|B(t)-I\|_2^2\right).$$

□

E. Additional Results on FLUX

We further evaluate OrthoFuse on the FLUX model to verify that our merging procedure can be applied to different model architectures. Figure 7 shows qualitative generations obtained after merging a style adapter and a concept adapter in FLUX. Each row corresponds to a distinct style-concept pair, while each column shows outputs for different text prompts applied to the same merged adapters.

The results demonstrate that OrthoFuse produces stable and coherent merges across a diverse set of style–concept combinations. Even under varying prompts, the merged adapters consistently preserve the underlying concept while expressing the intended style.

F. Necessity of Eigenvalue Rotation for High-Quality Merging

OrthoFuse method combines block-wise geodesic interpolation with *spectra restoration*, which can be considered as a specific eigenvalue rotation along the unit sphere, preserving orthogonality (see Section 4 for more details). While block-wise geodesics provide a natural and accurate approximation of the real local minimizing geodesic in practice, we observe that fusing $\mathcal{GS}$ orthogonal adapters with the help of block-wise geodesics only is insufficient for achieving high-quality semantic merging in diffusion models. Specifically, geodesic approximation via block-wise interpolation tends to drift away from the target concept and often fails to consistently align the style transformation across blocks.

Figure 8 illustrates this effect by comparing the merging trajectories obtained with block-wise geodesic approximation and our full OrthoFuse procedure. At intermediate interpolation levels – most clearly at $t = 0.6$ – the block-geodesic trajectory produces partially fused images where the transferred style is incomplete and the underlying concept begins to degrade. In contrast, OrthoFuse maintains both style fidelity and concept integrity, demonstrating that eigenvalue rotation plays a critical role in stabilizing the latent path and preventing semantic collapse.

All images in Figure 8 were generated with the prompt: "A <concept> <superclass> in jungle in <style> style."

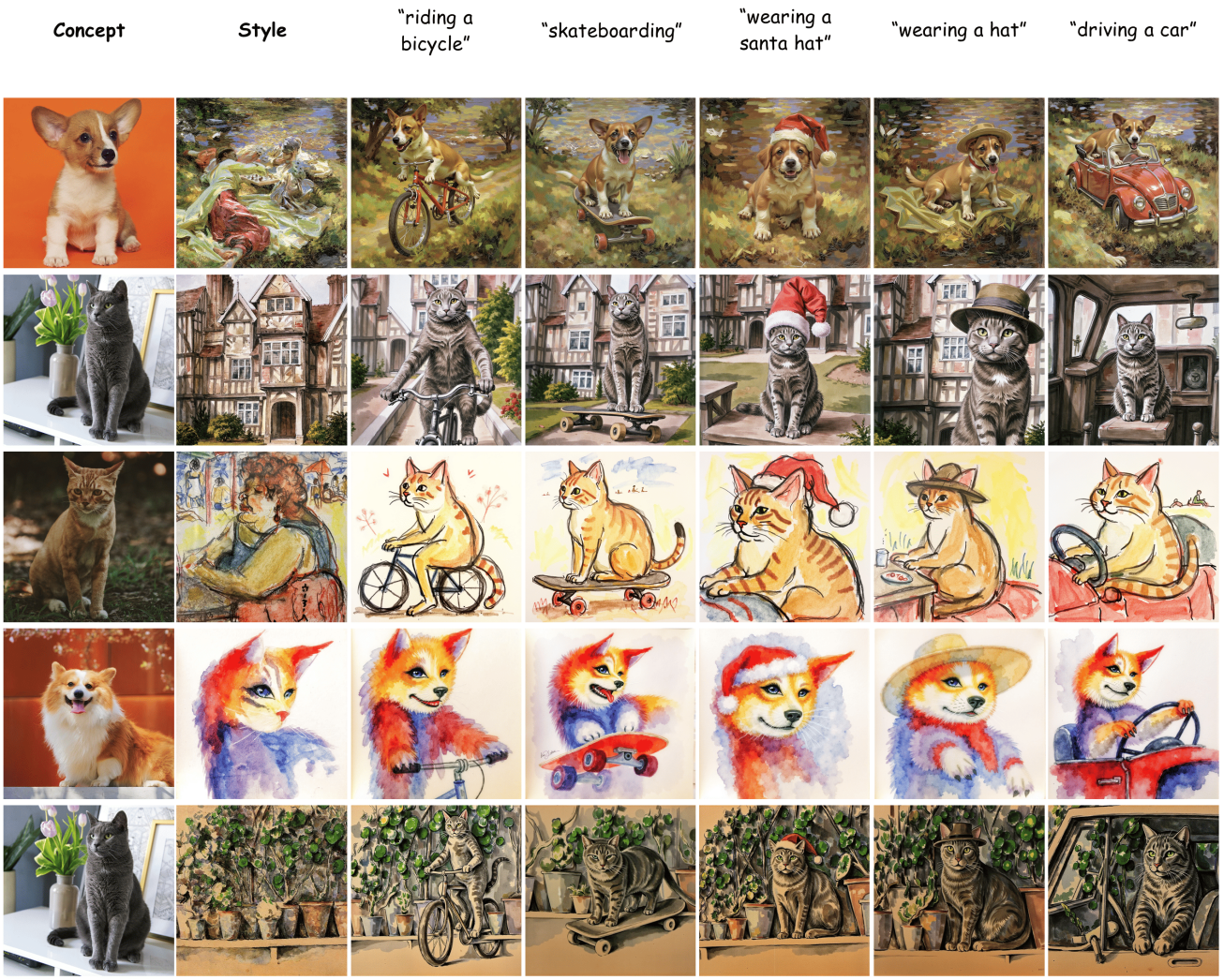


Figure 7. Qualitative results of **OrthoFuse** merging on the **FLUX** model. Each row shows generations produced from different prompts after merging a style adapter and a concept adapter. **OrthoFuse** maintains consistent concept preservation and style fidelity across prompts.

These results confirm that *spectra restoration* operation is not merely an auxiliary refinement but an essential operation for producing coherent and high-quality merges.

G. Additional Results on SDXL

We additionally provide extended qualitative results on the SDXL backbone to complement the evaluations in the main paper. Figure 9 presents generations obtained after merging a style adapter and a concept adapter within SDXL. In this visualization, each row corresponds to a different text prompt, while each column shows outputs for distinct style-concept adapter pairs applied to the same prompt.

Across all prompts and adapter configurations, OrthoFuse consistently achieves coherent style–concept integration, demonstrating strong concept preservation and stable expression of the intended style.

H. Ablation study on other merging methods

H.1. Low-rank adapter merging

To highlight the applicability of orthogonal fine-tuning and extend the experimental scale, we provide a similar adapter merging experiment but for a fixed-rank manifold: we train low-rank adapters with LoRA for style and concept and try to merge them as fixed-rank manifold elements. Assume that we aim to merge two low-rank matrices X_C and X_S which are low-rank weight updates for an arbitrary model layer:

$$X_C = U_C V_C^\top, \quad X_S = U_S V_S^\top, \tag{29}$$

where $U_C, U_S \in \mathbb{R}^{n\times r}, V_C, V_S \in \mathbb{R}^{m\times r}$. In the case of low-rank adapters, we aim to minimize the following ob-

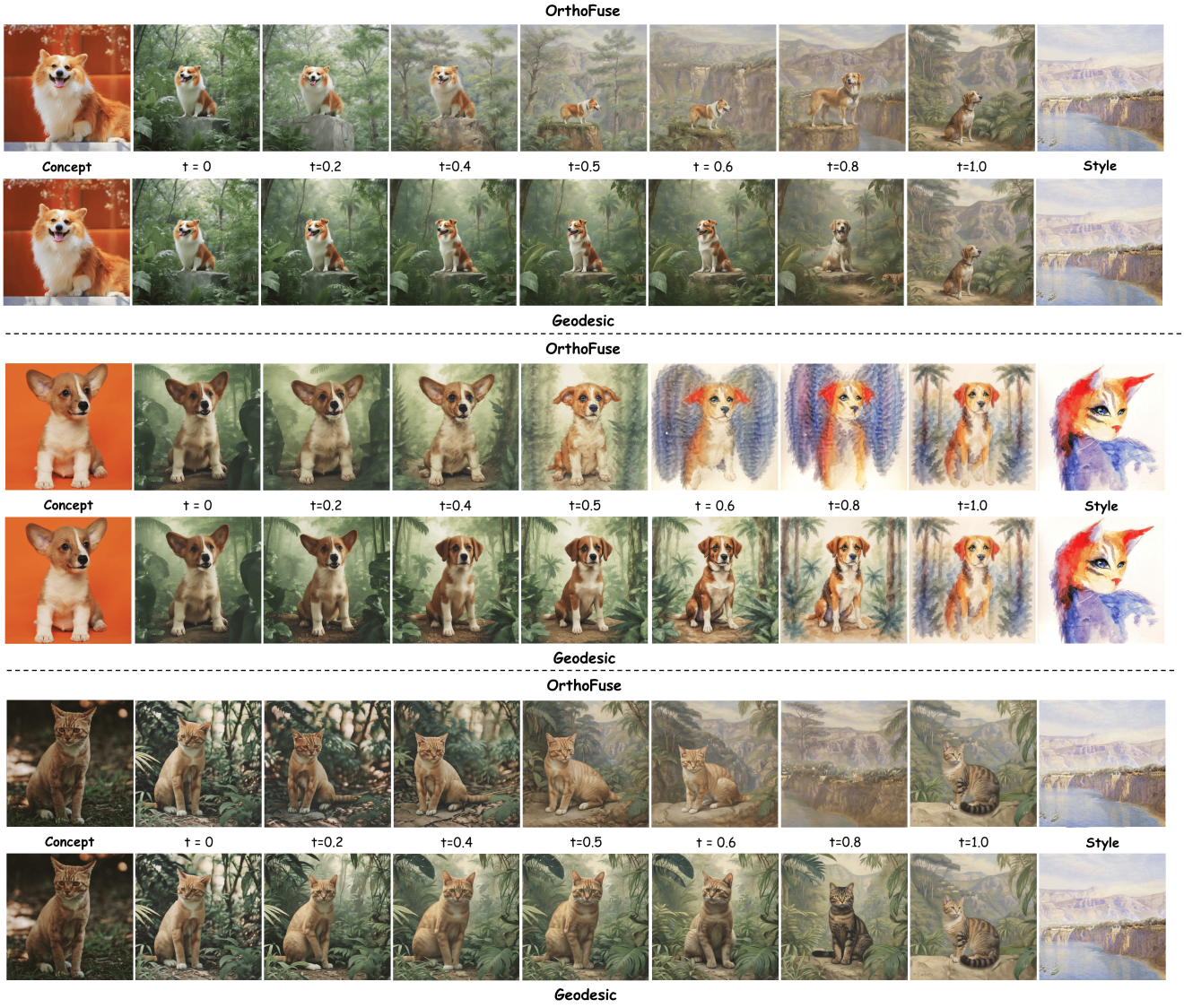


Figure 8. **Comparison of block-wise geodesic interpolation and OrthoFuse merging trajectories.** At $t = 0.6$, OrthoFuse achieves near-ideal style transfer while preserving the target concept. All images were generated with the prompt: "A <concept> < superclass> in jungle in <style> style".

jective: for $t \in [0, 1]$ we seek to optimize

$$t \cdot d^2_{\mathcal{M}_r}(X_t, X_S) + (1-t) \cdot d^2_{\mathcal{M}_r}(X_t, X_C) \to \min_{\mathrm{rk}(X_t)=r}, \quad (30)$$

where $d_{\mathcal{M}_r}(\cdot, \cdot)$ denotes the distance along the manifold e.g. the shortest curve between two points along the manifold. In our implementation, we replace the distance inside the manifold with the help of the Frobenius norm. Such a substitution is inspired by [18], which proposes certain theoretical guarantees on the closeness of such an approximation when the optimization is done with the change of the exact distance to its upper bound. Having replaced the manifold distance with the Frobenius norm, we obtain the following minimization:

$$t\|X_t - X_S\|^2_F + (1-t)\|X_t - X_C\|^2_F \to \min_{\mathrm{rk}(X_t)=r}. \quad (31)$$

This problem appears to be solved efficiently via ALS algorithm. Indeed, for the current low-rank approximation of $X_t = U_t V_t^\top$ we are able to alternately update its skeleton factors with short recurrent formulas:

- **V-step:** QR-decomposing $U = Q_U R$ and $X_t = Q_U \hat{V}$, we rewrite the task to the following one:

$$\|\hat{V}\|^2_F - 2\langle \hat{V}^\top, Q_U(tX_S + (1-t)X_C)\rangle \to \min_{\hat{V}} \quad (32)$$

Figure 9. Qualitative results of **OrthoFuse** merging on **SDXL**. Rows correspond to different prompts; columns show generations obtained from different style–concept adapter pairs. OrthoFuse yields coherent style–concept merges across both prompts and adapter combinations.

Taking the gradient by $\hat{V}$, it gives us the update for V:

$$\begin{aligned} &2\hat{V} - 2\left(Q_U^\top \left(tX_C + (1-t)X_S\right)\right)^\top = 0 \Rightarrow \\ \Rightarrow &\hat{V} = \left(tX_C + (1-t)X_S\right)^\top Q_U \end{aligned} \tag{33}$$

- **U-step**: in a similar to V-step manner, one can obtain the following update rule for U: considering QR-decomposition of $V = Q_V R$ and $X_t = \hat{U} Q_V^\top$ we need to

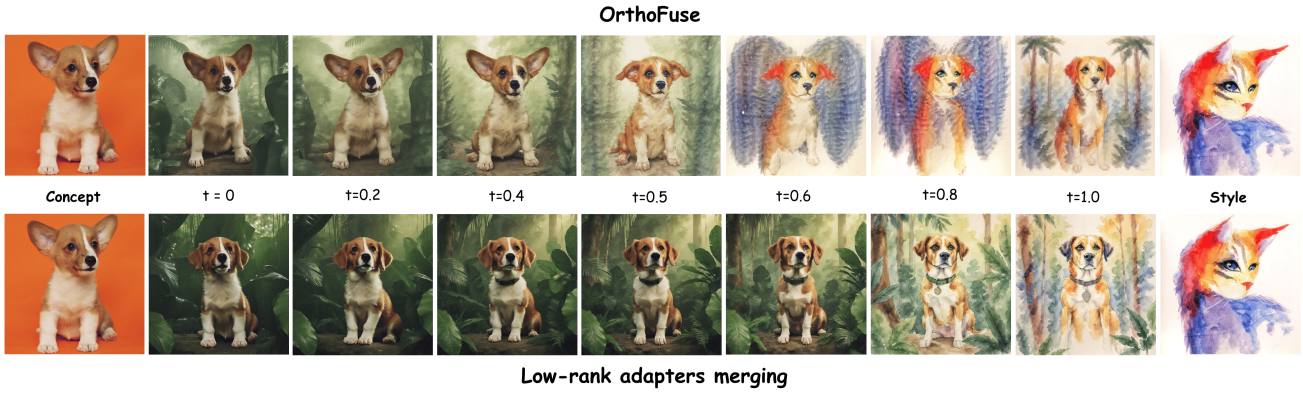


Figure 10. **Comparison of OrthoFuse (orthogonal adapters merging) with merging low-rank (LoRA) adapters.** Merging low-rank adapters results in noticeably weaker performance compared to OrthoFuse, even when both approaches are tuned to use approximately the same number of trainable parameters. We attribute this gap to the scale mismatch inherent to low-rank adapters, which makes their merging substantially more difficult. All images were generated with the prompt: "a <concept> dog in the jungle in <style> style".

solve the same optimization problem

$$\|\hat{U}\|_F^2 - 2\langle \hat{U}^\top, Q_V(tX_S + (1-t)X_C)\rangle \to \min_{\hat{U}}, \quad (34)$$

from which we immediately obtain

$$\hat{U} = (tX_C + (1-t)X_S)^\top Q_V. \quad (35)$$

It is worth mentioning that this problem can be easily generalized to the case of several low-rank adapters. In this case, in V-step and U-step one need to replace the term $(tX_C - (1-t)X_S)$ with the weighted sum of the corresponding low-rank adapters.

To validate this approach, in Figure 10 we report the empirical performance of the proposed merging method and compare it with OrthoFuse side by side. It can be observed that merging method applied to low-rank adapters performs worse than for $\mathcal{GS}$ orthogonal adapters failing to preserve concept pattern and style fidelity. To make the comparison fair, both methods were tuned using approximately the same number of parameters in corresponding parameter-efficient adapters.

H.2. Orthogonal adapter merging via multiplication

In order to additionally explain the motivation to take into account the geometry of the $\mathcal{GS}$ orthogonal manifold, we try the evaluate the most trivial way to merge orthogonal adapters together by multiplying their weight updates. On Figure 11 we report images obtained by multiplying orthogonal weight updates of the concept and style respectively. It can be seen that such an approach to fuse orthogonal adapters fails to preserve both style and concept patterns, which shows that for orthogonal adapters we need a more complicated approach which explicitly treats the structure of both orthogonal adapters.

I. Ablation study of Fusion Parameters

In this appendix, we present ablation studies analyzing the impact of two key fusion parameters, η_0 and t, on the performance of our proposed method.

I.1. Ablation of η_0

Figure 12 illustrates the results of the ablation study for the fusion parameter η_0:

- When $\eta_0 = 0$, the merged adapter collapses to the identity matrix, leading to no concept or style blending.
- At $\eta_0 = 1$, the model reduces to block-wise geodesic interpolation.
- The optimal performance is observed around ($\eta_0 \approx 2$), while larger values tend to degrade performance.

This analysis emphasizes the importance of appropriately selecting the fusion parameter η_0 to achieve the desired balance between concept and style.

I.2. Ablation of t

Figure 13 presents the results of the ablation study for the fusion parameter t:

- At $t = 0$, the method maximizes image similarity and minimizes style similarity.
- As t increases, image similarity decreases while style similarity increases, with the maximum style similarity achieved at $t \approx 0.8$.
- Notably, the stylistic effects exhibited at $t = 0.8$ are stronger than those observed at $t = 1$. This is because, for $t < 1$, the eigenvalue transformation is applied, which can amplify the stylistic components. When $t = 1$, this transformation is disabled to recover the original style adapter from (7), which can make the result at $t = 0.8$ appear stylistically stronger by comparison.

Figure 11. **Direct Merging via Multiplication.** The result of directly merging orthogonal adapters, accomplished through the multiplication of two $\mathcal{GS}$ orthogonal matrices, exhibits limitations in style preservation, struggles to maintain color consistency, and has a negative impact on concept fidelity.

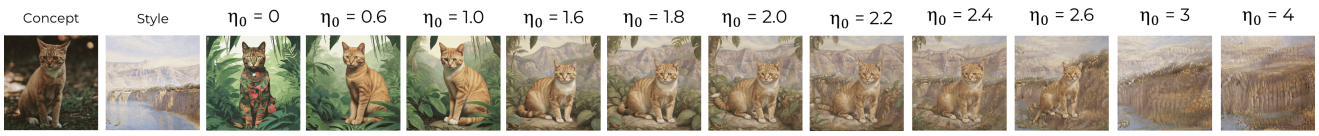


Figure 12. Ablation of the fusion parameter η_0.

This behavior indicates the trade-off between image and style similarity, underscoring the significance of fine-tuning parameter t for optimal performance.

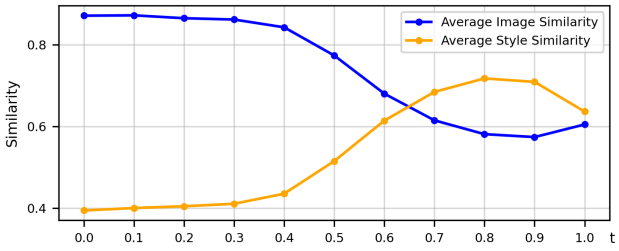


Figure 13. Ablation of the fusion parameter t.

J. User Study

In this appendix, we provide further details about the user study conducted to evaluate the effectiveness of our proposed method compared to K-LoRA and ZipLoRA. To address the limitations of automatic metrics in assessing concept–style trade-offs, we designed a user study involving 65 participants. This study resulted in 1,460 pairwise comparisons across all images used in the evaluation.

Participants were asked to compare images generated by different methods applied to the same concept–style pair and respond to the following questions:

Q1: Which image better captures the features of the style? Evaluate whether the style is recognizable through visual characteristics (colors, textures, brush strokes, lines, etc.); can we say that the concept is genuinely represented in this style, rather than just slightly altered?

Q2: Which image better preserves the concept? Assess how well the original object (concept) is maintained; is it recognizable (shape, proportions, structure), and are important details retained? Please disregard any changes in pose.

Q3: Which method, in general, handled the task of style transfer to the concept better? Assess the overall result of the style transfer:

- Does it create the impression that the concept is naturally executed in the given style?
- How well does the style harmonize with the object?

If you do not see a difference regarding any question or are uncertain about your choice, please select ”not sure.”

K. OrthoFuse Implementation Details

This section provides implementation details of the OrthoFuse algorithm used to construct the fused adapter $A(t)$ from independently trained concept and style adapters.

As described in Section 3.2, both adapters are represented via block structures. In algorithms below we denote concept and style corresponding weight matrices $(D_C^{(i)}, D_S^{(i)})$. Importantly, $(D_C^{(i)}, D_S^{(i)})$ are weight matrices which are used to build skew-hermitian matrices for a subsequent Cayley transform application.

All merging operations are then performed independently on each corresponding pair of blocks.

K.1. Full OrthoFuse: Geodesic Block Interpolation

The full OrthoFuse method performs interpolation along the geodesic in the orthogonal group for each block.

For every pair $\big(B_C^{(i)}, B_S^{(i)}\big)$ we compute:

$$\widetilde{B}^{(i)}(t) = \text{Geodesic}\big(B_C^{(i)}, B_S^{(i)}, t\big). \tag{36}$$

In practice, the geodesic is computed via:

1. Conversion to skew-symmetric generators using the Cayley parameterization;
2. Spectral decomposition of $B_S^\top B_C$;
3. Logarithmic interpolation in the Lie algebra;
4. Exponential map back to the orthogonal group

The corresponding pseudocode for a single block is shown below.

Algorithm 1 OrthoFuse merging

Require: D_C, D_S.
1: $K_C = \frac{D_C - D_C^\top}{2}$; $K_S = \frac{D_S - D_S^\top}{2}$;
2: $B_C = \texttt{torch.linalg.solve}((I - K_C)(I + K_C))$;
$B_S = \texttt{torch.linalg.solve}((I - K_S)(I + K_S))$;
3: $\Lambda, U = \texttt{torch.linalg.eig}(B_S^\top B_C)$;
4: $\Lambda_{\log} = \log(\Lambda).\texttt{imag} \cdot i$;
5: $B_t = B_C \texttt{torch.linalg.matrix\_exp}(-t \cdot U \Lambda_{\log} U^*).\texttt{real}$;
6: **return** B_t.

The postprocessing step is defined as follows.

Algorithm 2 OrthoFuse postprocess

Require: B_t.
1: $\eta = 1 + 4t(1-t)$
2: $Q = \eta B_t / 2$;
3: $Q^{skew} = \frac{Q - Q^\top}{2}$;
4: $Q = \texttt{torch.linalg.solve}((I - Q^{skew})(I + Q^{skew}))$;
5: **return** Q.

Overall, the full OrthoFuse procedure is defined as: `OrthoFuse = OrthoFuseMerging + OrthoFusePostprocess`.

K.2. Accelerated OrthoFuse (Merge Inside Cayley Space)

We also implement a computationally efficient approximation.

Instead of performing geodesic interpolation in $\mathrm{O}(k)$, we interpolate directly in the Cayley parameter space:

$$D_{merge}^{(i)} = tD_C^{(i)} + (1-t)D_S^{(i)} \tag{37}$$

The merged skew-symmetric block is then mapped to the orthogonal group via the Cayley transform.

Algorithm 3 OrthoFuse: merge inside Cayley space

Require: D_C, D_S.
1: $D_{merge} = tD_C + (1-t)D_S$
2: $K_{merge} = \frac{D_{merge} - D_{merge}^\top}{2}$;
3: $B_t = \texttt{torch.linalg.solve}((I - K_{merge})(I + K_{merge}))$;
4: **return** B_t.

The full accelerated pipeline is therefore:
`FastOrthoFuse = MergeInsideCayleySpace + OrthoFusePostprocess`.

K.3. Theoretical Justification of Accelerated OrthoFuse

First, we show that the matrix logarithm is correctly defined.

Lemma 3. *Assume orthogonal matrices $B_S, B_C \in \mathrm{SO}(n)$ satisfy $\|B_S - I\|_2 \le \varepsilon < 1$ and $\|B_C - I\|_2 \le \varepsilon < 1$. Then the matrix logarithm $\log(B_S^\top B_C)$ is well-defined.*

Proof. Using the triangle inequality and submultiplicativity, we bound the spectral norm of the difference:

$$\|B_S^\top B_C - I\|_2 \le \|B_S^\top - I\|_2 + \|B_C - I\|_2 \le 2\varepsilon.$$

Since $B_S^\top B_C$ is orthogonal, its eigenvalues lie on the unit circle. If -1 were an eigenvalue, the distance to the identity I would be at least $|-1-1| = 2$. Hence, whenever $2\varepsilon < 2$, the matrix $B_S^\top B_C$ has no eigenvalues equal to -1. □

Now we make use of the following auxiliary lemma.

Lemma 4. *For sufficiently small matrices X and Y with $\|X\|_2, \|Y\|_2 = \mathcal{O}(\varepsilon)$, the matrix logarithm of their exponential product is given by:*

$$\log(\exp(X)\exp(Y)) = X + Y + \frac{1}{2}[X, Y] + \mathcal{O}(\varepsilon^3)$$

Proof. This follows from well-known (see, for example, [16]) Baker-Campbell-Hausdorff formula. By submultiplicativity, the norm of the commutator satisfies $\|[X, Y]\|_2 \le 2\|X\|_2\|Y\|_2 = \mathcal{O}(\varepsilon^2)$. Consequently, nested commutators such as $[X, [X, Y]]$ inherently possess a higher order of smallness $\mathcal{O}(\varepsilon^3)$. Similarly arguing by induction, it is straightforward to establish that each subsequent nesting of the commutator increases the order of smallness. □

Finally, we prove that geodesics can be approximated by a connection in the space of skew-symmetric matrices.

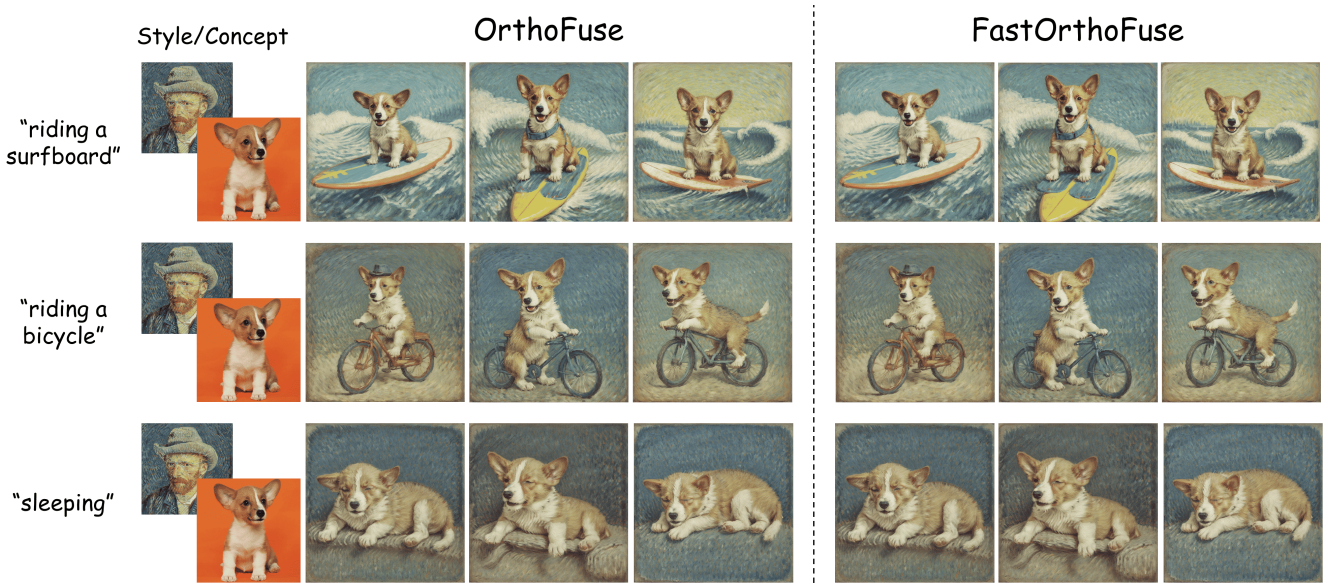


Figure 14. **Comparative analysis of OrthoFuse and its accelerated version**. Using identical concept and style adapters and the same fusion parameter t, both methods produce visually indistinguishable results. The accelerated version removes the eigendecomposition step while preserving identity and style fidelity. We note, however, that the two methods are not strictly identical; for example, in the first row, the dog's right paw in the OrthoFuse result is not fully placed on the surfboard, whereas in FastOrthoFuse it is.

Proposition 11. *Let $B_C, B_S \in \mathrm{SO}(n)$ be orthogonal matrices parameterized by skew-symmetric matrices $D_C, D_S \in \mathfrak{so}(n)$ via the Cayley transform: $B_C = \mathrm{Cayley}(D_C)$ and $B_S = \mathrm{Cayley}(D_S)$, where $\|D_C\|_2, \|D_S\|_2 = \mathcal{O}(\varepsilon)$. The Cayley transform of their linearly interpolated generators approximates the exact Riemannian geodesic $B(t) = B_C \exp(-t \log(B_S^\top B_C))$ up to a third-order error:*

$$B(t) = \mathrm{Cayley}((1-t)D_C + tD_S) + \mathcal{O}(\varepsilon^3)$$

Proof. Recall that the Cayley transform matches the matrix exponential up to the second order: $\mathrm{Cayley}(K) = \exp(K) + \mathcal{O}(\varepsilon^3)$. We rewrite our endpoints as $B_C = \exp(D_C) + \mathcal{O}(\varepsilon^3)$ and $B_S^\top = \mathrm{Cayley}(-D_S) = \exp(-D_S) + \mathcal{O}(\varepsilon^3)$. Applying Lemma 4, we approximate the logarithm term:

$$\begin{aligned}\log(B_S^\top B_C) &= \log(\exp(-D_S)\exp(D_C)) \\ &= D_C - D_S - \frac{1}{2}[D_S, D_C] + \mathcal{O}(\varepsilon^3)\end{aligned}$$

Substituting this into the geodesic equation yields $B(t) = \exp(D_C)\exp(V) + \mathcal{O}(\varepsilon^3)$, where the exponent is defined as $V = -t(D_C - D_S) + \frac{t}{2}[D_S, D_C]$. We apply Lemma 4 a second time to combine these into a single exponential $\exp(Z)$, where $Z = D_C + V + \frac{1}{2}[D_C, V]$.

Given that $[D_C, V] = \frac{1}{2}[D_C, -t(D_C - D_S)] + \mathcal{O}(\varepsilon^3)$, we expand Z:

$$\begin{aligned}Z = D_C - t(D_C - D_S) + \frac{t}{2}[D_S, D_C] \\ + \frac{1}{2}[D_C, -t(D_C - D_S)]\end{aligned}$$

Due to the anti-symmetry of the Lie bracket, the second-order commutators perfectly cancel each other out:

$$\frac{1}{2}[D_C, -t(D_C - D_S)] = \frac{t}{2}[D_C, D_S] = -\frac{t}{2}[D_S, D_C]$$

This exact cancellation reduces the exponent to $Z = (1-t)D_C + tD_S$. Therefore, the geodesic simplifies to $B(t) = \exp((1-t)D_C + tD_S) + \mathcal{O}(\varepsilon^3)$. Applying the Padé equivalence $\exp(Z) = \mathrm{Cayley}(Z) + \mathcal{O}(\varepsilon^3)$ once more concludes the proof. □

Another way to think about this is as follows. After training the adapters using skew-symmetric matrices, we obtain their final representations and then apply the Cayley transform (a retraction) to obtain orthogonal matrices. Thus, performing linear interpolation in the skew-symmetric parameter space corresponds to mixing weights — a concept reminiscent of task arithmetic (see, e.g., [12]).

K.4. Computational Considerations

The computational bottleneck of the full OrthoFuse algorithm is the eigendecomposition step:

$$\Lambda, U = \texttt{torch.linalg.eigh}(Q_S^\top Q_C). \tag{38}$$

For the matrix sizes used in our adapters, this operation accounts for approximately **90% of the total merging time**. In PyTorch, this routine is not efficiently parallelized in our setting and dominates the wall-clock runtime.

The accelerated variant completely removes the eigendecomposition. All remaining operations (matrix multiplications, linear solves, and matrix exponentials) are efficiently parallelized, leading to a substantial speedup. In practice, the fast version performs adapter merging in under one second while nonaccelerated version works in 90 seconds.

K.5. Empirical Observation

Despite the simplification, the accelerated version produces images that are visually almost indistinguishable from those obtained using the full geodesic interpolation (see Figure 14). In our experiments, we observe no meaningful degradation in identity preservation or style transfer quality, while achieving a significant reduction in computational cost.

L. Limitations

While OrthoFuse is training-free, it requires adapters to be in $\mathcal{GS}$-orthogonal form. Most community adapters are standard LoRAs; applying our method directly to them would need a projection, which risks losing the information encoded in the original LoRA weights. Extending our fusion to LoRA weights is a promising future direction.