Appendix

Operational properties of dual quaternion (DualE)

Conjugate: The conjugate of a dual quaternion Q is defined as: $Q^* = (a_0, -a_1, -a_2, -a_3, b_0, -b_1, -b_2, -b_3)$.
Norm: The norm $\|Q\|$ of a dual quaternion Q is a dual scalar and it is defined as: $\sqrt{(a_0+\epsilon b_0)^2 + (a_1+\epsilon b_1)^2 + (a_2+\epsilon b_2)^2 + (a_3+\epsilon b_3)^2}$.
Unit dual quaternion : A dual quaternion $Q = a + \epsilon b$ is unit if and only if $\|Q\| = 1 \ \wedge \ a \cdot b = 0$.
Inner Product: The dual quaternion inner product $\langle Q_1, Q_2 \rangle$ between $Q_1 = (a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3)$ and $Q_2 = (c_0, c_1, c_2, c_3, d_0, d_1, d_2, d_3)$ is obtained by taking the inner product of the dual quaternions between the corresponding real and dual parts, respectively, and sum the four parts:

$$\begin{aligned} Q_1 \bullet Q_2 =& \langle a_0, c_0 \rangle + \langle b_0, d_0 \rangle + \langle a_1, c_1 \rangle + \langle b_1, d_1 \rangle \\ &+ \langle a_2, c_2 \rangle + \langle b_2, d_2 \rangle + \langle a_3, c_3 \rangle + \langle b_3, d_3 \rangle \end{aligned} \tag{13}$$

Dual Quaternion Multiplication: Let $a_h = a_0 + \epsilon b_0$, $b_h = a_1 + \epsilon b_1$, $c_h = a_2 + \epsilon b_2$, $d_h = a_3 + \epsilon b_3$, $p = c_0 + \epsilon c_0$, $q = c_1 + \epsilon d_1$,$u = c_2 + \epsilon d_2$, $v = c_3 + \epsilon d_3$. The dual quaternion multiplication between Q_1 and Q_2 can be divided into two steps. First multiply the generalized quaternions according to the settings above, and then multiply the real and dual parts of the dual quaternions in each part, as follows:

$$\begin{aligned} Q_1 \underline{\otimes} Q_2 =& (a_h p - b_h q - c_h u - d_h v) \\ &+ (a_h q + b_h p + c_h v - d_h u)\,\mathbf{i} \\ &+ (a_h u - b_h v + c_h p + d_h q)\,\mathbf{j} \\ &+ (a_h v + b_h u - c_h q + d_h p)\,\mathbf{k} \end{aligned} \tag{14}$$

Proof and more details for Eq.(4): The quaternion σ is $r + \frac{\epsilon}{2}\boldsymbol{t}r$, and its conjugate is $\sigma^* = r^* + \frac{\epsilon}{2}(\boldsymbol{t}r)^*$. Next, we derive the following formula:

$$\begin{aligned} \sigma \otimes \sigma^* &= \left(r + \frac{\epsilon}{2} tr\right)\left(r^* + \frac{\epsilon}{2}(tr)^*\right) \\ &= \left(r + \frac{\epsilon}{2} tr\right)\left(r^* + \frac{\epsilon}{2} r^* t^*\right) \\ &= r r^* + \frac{\epsilon}{2}\left(r r^* t^* + t r r^*\right) \\ &= 1 + \frac{\epsilon}{2}\left(t^* + t\right) \\ &= 1 \end{aligned} \tag{15}$$

where the last step made use of the fact that $\boldsymbol{t}$ is a pure quaternion with $\boldsymbol{t}^* = -\boldsymbol{t}$.

For the multiplication of dual quaternions σ_1 and σ_2, right-multiplication by a unit dual quaternion σ_1 is the combination of right-isoclinic rotation[2] and translation on dual quaternion σ_2. And left-multiplication remains the similar geometrical meaning as right-multiplication.

Dual quaterion space. We compare DualE with TransE, HolE (Nickel, Rosasco, and Poggio 2016) and DistMult (Yang et al. 2015), which are used to embed in Euclidean space. We also compare DualE with ComplEx, RotatE and QuatE, which run in the complex or hypercomplex space. Recalling the form of dual quaternion, if the dual part is zero, it can degenerate into a quaternion; if the last two imaginary parts of the quaternion are zero, it can degenerate into a complex number. Combined with properties of the linear space, we can clearly see that DualE resides in the dual quaternion space and the other embedding spaces above are its subspaces.

Details for Table 1. As (Sun et al. 2019) mentions, the rotation family can completely model the key patterns (symmetry, antisymmetry, inversion and composition), but cannot model multiple relations pattern. For translation family, TransE cannot model symmetry and multiple relations patterns; TranH, TranR, etc. cannot model inversion and composition patterns. So on the whole, the translation family also cannot completely model the key patterns and multiple relations pattern.

[2]Isoclinic rotation represents a transformation in the four-dimensional space corresponding to the three-dimensional Euclidean transformation.

Algorithm 1

We remodel the initialization algorithm in (Parcollet et al. 2018) tailored for quaternion-valued networks to accelerate the calculation and convergence of the model effectively and convergence (Glorot and Bengio 2010) for parameters initilaization of DualE.

As Eq.(4) shown, the dual quaternion σ has the following form:

$$\begin{aligned} \sigma =& r + \frac{\epsilon}{2} tr \\ =& \cos\frac{\theta}{2} + \hat{u}\sin\frac{\theta}{2} + \frac{\epsilon}{2}(-\sin\frac{\theta}{2}(t-\hat{u}) + \cos\frac{\theta}{2}t \\ &+ \sin\frac{\theta}{2} t \times \hat{u}) \\ =& w_{real} + w_i \boldsymbol{i} + w_j \boldsymbol{k} + w_k \boldsymbol{k} + \frac{-t_i w_i - t_j w_j - t_k w_k}{2}\epsilon \\ &+ \frac{t_i w_{real} + t_j w_k - t_k w_j}{2}\epsilon \boldsymbol{i} + \frac{-t_i w_k + t_j w_{real} + t_k w_i}{2}\epsilon \boldsymbol{j} \\ &+ \frac{t_i w_j - t_j w_i + t_k w_{real}}{2}\epsilon \boldsymbol{k} \end{aligned} \tag{16}$$

For the entities and relations, we initializate $u = w_{real} + w_i \boldsymbol{i} + w_j \boldsymbol{j} + w_k \boldsymbol{k}$.

$$\begin{aligned} &w_{\text{real}} = \varphi\cos(\frac{\theta}{2}), w_i = \varphi Q^{\diamond}_{img_i}\sin(\frac{\theta}{2}), \\ &w_j = \varphi Q^{\diamond}_{img_j}\sin(\frac{\theta}{2}), w_k = \varphi Q^{\diamond}_{img_k}\sin(\frac{\theta}{2}) \end{aligned} \tag{17}$$

where w_{real}, w_i, w_j, w_k denote the scalar and imaginary coefficients, respectively. $\frac{\theta}{2}$ is randomly generated from the interval $[-\pi, \pi]$. $Q^{\diamond}_{img}$ is normalized quaternion, whose scalar part is zero. φ is randomly generated from the interval $[-\frac{1}{\sqrt{2k}}, \frac{1}{\sqrt{2k}}]$. $Q^{\diamond}_{img}$ and t are generated in the similar way.

Table 5: Scoring functions of state-of-the-art knowledge graph embedding models .

Model	Scoring Function	Parameters	$\mathcal{O}_{time}$
TransE	$\|(Q_h + W_r) - Q_t\|$	$Q_h, W_r, Q_t \in \mathbb{R}^k$	$\mathcal{O}(k)$
Hole	$\langle W_r, Q_h \star Q_t \rangle$	$Q_h, W_r, Q_t \in \mathbb{R}^k$	$\mathcal{O}(k \log(k))$
DistMult	$\langle W_r, Q_h, Q_t \rangle$	$Q_h, W_r, Q_t \in \mathbb{R}^k$	$\mathcal{O}(k)$
ComplEx	$Re(\langle W_r, Q_h, Q_t \rangle)$	$Q_h, W_r, Q_t \in \mathbb{C}^k$	$\mathcal{O}(k)$
RotatE	$\|Q_h \circ W_r - Q_t\|$	$Q_h, W_r, Q_t \in \mathbb{C}^k, \|W_{ri}\| = 1$	$\mathcal{O}(k)$
TourE	$min_{(x,y)\in([Q_h]+[Q_h])\times[W_r]}\|x - y\|$	$[Q_h], [W_r], [Q_t] \in \mathbb{T}^k$	$\mathcal{O}(k)$
QuatE	$Q_h \otimes W_r^{\triangleleft} \cdot Q_t$	$Q_h, W_r, Q_t \in \mathbb{H}^k$	$\mathcal{O}(k)$
DualE	$\langle Q_h \underline{\otimes} W_r^{\diamond}, Q_t \rangle$	$Q_h, W_r, Q_t \in \mathbb{H}_d^k$	$\mathcal{O}(k)$

Algorithm 1 Quaternion-valued weight initialization

procedure Qinit(W,n_{in},n_{out})
  $\sigma \leftarrow \frac{1}{\sqrt{2(n_{in})+n_{out}}}$
  for w *in* W **do**
    $\theta \leftarrow rand(-\pi, \pi)$
    $\varphi \leftarrow rand(-\sigma, \sigma)$
    $x, y, z = rand(0, 1), p, q, r = rand(0, 1)$
    $q_{imag} \leftarrow Quaternion(0, x, y, z)$
    $t_{origin} \leftarrow Quaternion(0, p, q, r)$
    $q_{imag}^{\diamond} \leftarrow \frac{q_{imag}}{\sqrt{x^2+y^2+z^2}}, t \leftarrow \frac{t_{origin}}{\sqrt{p^2+q^2+u^2}}$
    $w_r \leftarrow \varphi \times \cos(\frac{\theta}{2})$
    $w_i \leftarrow \varphi \times q_{imag_i}^{\diamond} \times \sin(\frac{\theta}{2})$
    $w_j \leftarrow \varphi \times q_{imag_j}^{\diamond} \times \sin(\frac{\theta}{2})$
    $w_k \leftarrow \varphi \times q_{imag_k}^{\diamond} \times \sin(\frac{\theta}{2})$
    $w \leftarrow Quaternion(w_r, w_i, w_j, w_k)$

After some tests, we found that the initialization scheme is optional on these four datasets, random initialization can get the same performance. This initialization scheme might be useful for the case which needs fewer epochs.

Proof of Lemma 1

Proof *if $r(x, y)$ and $r(y, x)$ hold, we have*

$$\mathbf{y} = \mathbf{r} \circ \mathbf{x} \wedge \mathbf{x} = \mathbf{r} \circ \mathbf{y} \Rightarrow \mathbf{r} \circ \mathbf{r} = 1$$

Otherwise, if $r(x, y)$ and $\neg r(y, x)$ hold, we have

$$y = r \circ x \wedge x \neq r \circ y \Rightarrow r \circ r \neq 1$$

The symmetry property of DualE can be proved by setting the imaginary parts of real part and dual part of W_r to zero.

In order to prove the antisymmetry pattern, we need to prove the following inequality when imaginary parts of real part and dual part are nonzero:

$$Q_h \underline{\otimes} W_r^{\diamond} \bullet Q_t \neq Q_t \underline{\otimes} W_r^{\diamond} \bullet Q_h \tag{18}$$

And we let

$$\begin{aligned} a_h &= a_0 + \epsilon b_0, b_h = a_1 + \epsilon b_1 \\ c_h &= a_2 + \epsilon b_2, d_h = a_3 + \epsilon b_3 \\ p &= c_0 + \epsilon d_0, q = c_1 + \epsilon d_1 \\ u &= c_2 + \epsilon d_2, v = c_3 + \epsilon d_3 \\ a_t &= e_0 + \epsilon f_0, b_t = e_1 + \epsilon f_1 \\ c_t &= e_2 + \epsilon f_2, d_t = e_3 + \epsilon f_3 \end{aligned}$$

$$\begin{aligned} & Q_h \underline{\otimes} W_r^{\diamond} \bullet Q_t \\ =& [(a_h \bullet p - b_h \bullet q - c_h \bullet u - d_h \bullet v) + \\ & (a_h \bullet q + b_h \bullet p + c_h \bullet v - d_h \bullet u)\,\mathbf{i} + \\ & (a_h \bullet u - b_h \bullet v + c_h \bullet p + d_h \bullet q)\,\mathbf{j} + \\ & (a_h \bullet v + b_h \bullet u - c_h \bullet q + d_h \bullet p)\,\mathbf{k}] \\ & \bullet (a_t + b_t\mathbf{i} + c_t\mathbf{j} + d_t\mathbf{k}) \\ =& (a_h \bullet p - b_h \bullet q - c_h \bullet u - d_h \bullet v) \bullet a_t + \\ & (a_h \bullet q + b_h \bullet p + c_h \bullet v - d_h \bullet u) \bullet b_t + \\ & (a_h \bullet u - b_h \bullet v + c_h \bullet p + d_h \bullet q) \bullet c_t + \\ & (a_h \bullet v + b_h \bullet u - c_h \bullet q + d_h \bullet p) \bullet d_t \\ =& \langle a_h, p, a_t \rangle - \langle b_h, q, a_t \rangle - \langle c_h, u, a_t \rangle - \langle d_h, v, a_t \rangle + \\ & \langle a_h, q, b_t \rangle + \langle b_h, p, b_t \rangle + \langle c_h, v, b_t \rangle - \langle d_h, u, b_t \rangle + \\ & \langle a_h, u, c_t \rangle - \langle b_h, v, c_t \rangle + \langle c_h, p, c_t \rangle + \langle d_h, q, c_t \rangle + \\ & \langle a_h, v, d_t \rangle + \langle b_h, u, d_t \rangle - \langle c_h, q, d_t \rangle + \langle d_h, p, d_t \rangle \end{aligned} \tag{19}$$

And then, we expand the right term:

$$\begin{aligned} & Q_t \underline{\otimes} W_r^{\diamond} \bullet Q_h \\ =& [(a_t \bullet p - b_t \bullet q - c_t \bullet u - d_t \bullet v) + \\ & (a_t \bullet q + b_t \bullet p + c_t \bullet v - d_t \bullet u)\,\mathbf{i} + \\ & (a_t \bullet u - b_t \bullet v + c_t \bullet p + d_t \bullet q)\,\mathbf{j} + \\ & (a_t \bullet v + b_t \bullet u - c_t \bullet q + d_t \bullet p)\,\mathbf{k}] \\ & \bullet (a_h + b_h\mathbf{i} + c_h\mathbf{j} + d_h\mathbf{k}) \\ =& (a_t \bullet p - b_t \bullet q - c_t \bullet u - d_t \bullet v) \bullet a_h + \\ & (a_t \bullet q + b_t \bullet p + c_t \bullet v - d_t \bullet u) \bullet b_h + \\ & (a_t \bullet u - b_t \bullet v + c_t \bullet p + d_t \bullet q) \bullet c_h + \\ & (a_t \bullet v + b_t \bullet u - c_t \bullet q + d_t \bullet p) \bullet d_h \\ =& \langle a_t, p, a_h \rangle - \langle b_t, q, a_h \rangle - \langle c_t, u, a_h \rangle - \langle d_t, v, a_h \rangle + \\ & \langle a_t, q, b_h \rangle + \langle b_t, p, b_h \rangle + \langle c_t, v, b_h \rangle - \langle d_t, u, b_h \rangle + \\ & \langle a_t, u, c_h \rangle - \langle b_t, v, c_h \rangle + \langle c_t, p, c_h \rangle + \langle d_t, q, c_h \rangle + \\ & \langle a_t, v, d_h \rangle + \langle b_t, u, d_h \rangle - \langle c_t, q, d_h \rangle + \langle d_t, p, d_h \rangle \end{aligned}$$

By comparison, we can deduce that these two terms are not equal because some terms have different signs.

Proof of Lemma 2

Proof *if $r_1(x, y)$ and $r_2(y, x)$ hold, we have*

$$\mathbf{y} = \mathbf{r}_1 \circ \mathbf{x} \wedge \mathbf{x} = \mathbf{r}_2 \circ \mathbf{y} \Rightarrow \mathbf{r}_1 = \mathbf{r}_2^{-1}$$

We can utilize the conjugation of dual quaternions to prove the inversion pattern since conjugation of the dual quaternion is its own inverse. We need to prove that:

$$Q_h \underline{\otimes} W_r^{\diamond} \bullet Q_t = Q_t \underline{\otimes} \bar{W}_r^{\diamond} \bullet Q_h \tag{20}$$

Then we expand the right term:

$$\begin{aligned}
&Q_t \underline{\otimes} \bar{W}_r^{\diamond} \bullet Q_h \\
=&[(a_t \bullet p + b_t \bullet q + c_t \bullet u + d_t \bullet v) + \\
&(-a_t \bullet q + b_t \bullet p - c_t \bullet v + d_t \bullet u)\,\mathbf{i} + \\
&(-a_t \bullet u + b_t \bullet v + c_t \bullet p - d_t \bullet q)\,\mathbf{j} + \\
&(-a_t \bullet v - b_t \bullet u + c_t \bullet q + d_t \bullet p)\,\mathbf{k}] \\
&\bullet (a_h + b_h\mathbf{i} + c_h\mathbf{j} + d_h\mathbf{k}) \\
=&(a_t \bullet p - b_t \bullet q - c_t \bullet u - d_t \bullet v) \bullet a_h + \\
&(a_t \bullet q + b_t \bullet p + c_t \bullet v - d_t \bullet u) \bullet b_h + \\
&(a_t \bullet u - b_t \bullet v + c_t \bullet p + d_t \bullet q) \bullet c_h + \\
&(a_t \bullet v + b_t \bullet u - c_t \bullet q + d_t \bullet p) \bullet d_h \\
=&\langle a_t, p, a_h\rangle + \langle b_t, q, a_h\rangle + \langle c_t, u, a_h\rangle + \langle d_t, v, a_h\rangle - \\
&\langle a_t, q, b_h\rangle + \langle b_t, p, b_h\rangle - \langle c_t, v, b_h\rangle + \langle d_t, u, b_h\rangle - \\
&\langle a_t, u, c_h\rangle + \langle b_t, v, c_h\rangle + \langle c_t, p, c_h\rangle - \langle d_t, q, c_h\rangle - \\
&\langle a_t, v, d_h\rangle - \langle b_t, u, d_h\rangle + \langle c_t, q, d_h\rangle + \langle d_t, p, d_h\rangle
\end{aligned}$$

By comparison, we can see that the left and right are equal.

Proof of Lemma 3

Proof *if $r_1(x, z)$, $r_2(x, y)$ and $r_3(y, z)$ hold, we have*

$$\mathbf{z} = \mathbf{r}_1 \circ \mathbf{x} \wedge \mathbf{y} = \mathbf{r}_3 \circ \mathbf{x} \wedge \mathbf{z} = \mathbf{r}_2 \circ \mathbf{y} \Rightarrow \mathbf{r}_1 = \mathbf{r}_2 \circ \mathbf{r}_3$$

That is to say, for composition relations we can get that:

$$\begin{aligned}
&(Q_h \underline{\otimes} W_{r_2}^{\diamond}) \underline{\otimes} W_{r_3}^{\diamond} \bullet Q_t \\
=&Q_h \underline{\otimes} W_{r_2}^{\diamond} \underline{\otimes} W_{r_3}^{\diamond} \bullet Q_t \\
=&Q_h \underline{\otimes} (W_{r_2}^{\diamond} \underline{\otimes} W_{r_3}^{\diamond}) \bullet Q_t \\
=&Q_h \underline{\otimes} W_{r_1}^{\diamond} \bullet Q_t
\end{aligned}$$

Proof of Lemma 4

Proof *Recall Eq.(4), which gives the geometric meaning of dual quaternions. For a pair of head entity and tail entity in the embedding space, we can first give any r, which means that the head node first rotates an angle on a spherical surface, and then translate the rotated head node by $\mathbf{t}$ to reach the tail entity. In this transformation, any given angle has one corresponding to it. In this transformation, for any given r, there always is a $\mathbf{t}$ corresponding to it. So DualE can model multiple relations pattern.*

Proof of Lemma 5

Next, we prove that DualE is a more general representation of QuatE that can represent the rotation operation between entity relations.

Proof *ComplEx model scoring function:*

$$\begin{aligned}
\phi(r, s, o; \Theta) =& \operatorname{Re}\left(< w_r, e_s, \bar{e}_o >\right) \\
=& \operatorname{Re}\left(\sum_{k=1}^{K} w_{rk} e_{sk} \bar{e}_{ok}\right) \\
=& \langle \operatorname{Re}(w_r), \operatorname{Re}(e_s), \operatorname{Re}(e_o)\rangle + \\
&\langle \operatorname{Re}(w_r), \operatorname{Im}(e_s), \operatorname{Im}(e_o)\rangle + \\
&\langle \operatorname{Im}(w_r), \operatorname{Re}(e_s), \operatorname{Im}(e_o)\rangle - \\
&\langle \operatorname{Im}(w_r), \operatorname{Im}(e_s), \operatorname{Re}(e_o)\rangle
\end{aligned}$$

QuatE model scoring function with removing the normalization of the relational quaternion:

$$\begin{aligned}
&\phi(h, r, t) \\
=&Q_h \otimes W_r \cdot Q_t \\
=&(a_h + b_h\boldsymbol{i} + c_h\boldsymbol{j} + d_h\boldsymbol{k}) \otimes (a_r + b_r\boldsymbol{i} + c_r\boldsymbol{j} + d_r\boldsymbol{k}) \\
&\cdot (a_t + b_t\boldsymbol{i} + c_t\boldsymbol{j} + d_t\boldsymbol{k}) \\
&\xlongequal{unit_{\boldsymbol{i}\ and\ \boldsymbol{j}}=0} (a_h + b_h\mathbf{i}) \otimes (a_r + b_r\mathbf{i}) \\
&\cdot (a_t + b_t\mathbf{i}) \\
=&[(a_h \circ a_r - b_h \circ b_r) + (a_h \circ b_r + b_h \circ a_r)\,\mathbf{i}] \\
&\cdot (a_t + b_t\mathbf{i}) \\
=&\langle a_r, a_h, a_t\rangle + \langle a_r, b_h, b_t\rangle + \langle b_r, a_h, b_t\rangle - \\
&\langle b_r, b_h, a_t\rangle
\end{aligned}$$

If we set the coefficients of the imaginary units $\boldsymbol{j}$ and $\boldsymbol{k}$ to zero, we get complex embeddings as in ComplEx and the Hamilton product will also degrade to complex number multiplication. According to the two scoring functions above, we can find that QuatE recovers the form of ComplEx. If we remove the imaginary parts of all quaternions and remove the normalization step, the scoring function becomes $\phi(h, r, t) = \langle a_h, a_r, a_t\rangle$, degrading to DistMult in this case.

DualE model scoring function:

$$\begin{aligned}
&\phi(h, r, t) \\
=&Q_h \underline{\otimes} W_r \bullet Q_t \\
=&(a_0 + a_1\boldsymbol{i} + a_2\boldsymbol{j} + a_3\boldsymbol{k} + \epsilon(b_0 + b_1\boldsymbol{i} + b_2\boldsymbol{j} + b_3\boldsymbol{k})) \\
&\underline{\otimes}(c_0 + c_1\boldsymbol{i} + c_2\boldsymbol{j} + c_3\boldsymbol{k} + \epsilon(d_0 + d_1\boldsymbol{i} + d_2\boldsymbol{j} + d_3\boldsymbol{k})) \\
&\bullet (e_0 + e_1\boldsymbol{i} + e_2\boldsymbol{j} + e_3\boldsymbol{k} + \epsilon(f_0 + f_1\boldsymbol{i} + f_2\boldsymbol{j} + f_3\boldsymbol{k})) \\
&\xlongequal{unit_{\epsilon}=0} (a_h + b_h\boldsymbol{i} + c_h\boldsymbol{j} + d_h\boldsymbol{k}) \otimes \\
&(a_r + b_r\boldsymbol{i} + c_r\boldsymbol{j} + d_r\boldsymbol{k}) \cdot (a_t + b_t\boldsymbol{i} + c_t\boldsymbol{j} + d_t\boldsymbol{k})
\end{aligned}$$

We can see that when the dual part is zero, DualE degenerates into QuatE, and dual multiplication also degenerates into Hamiltonian product. Therefore, QuatE is a special case of DualE. Through the discussion above, we can see that the rotation family models are within the DualE model framework.

Proof of Lemma 6

Next, we prove that DualE is a more general representation of TransE that can represent the translation operations between entity relations. That is, QuatE family models are special form of DualE.

Definition 5 *For the translation family models, if only h is projected, it is called single projection.*

Definition 6 *For the translation family models, if h and t are projected in the same hyperplane or space, it is called double homogeneous projection.*

Definition 7 *For the translation family models, if h and t are projected in different hyperplanes or spaces, it is called double different projection.*

Proof *Recalling Eq.(16), let $\theta = 0$, then we end up with $\sigma = 1 + \frac{\epsilon}{2}\boldsymbol{t}$. We can know that it only represents the translation. Set the translation distance $t = r$, and we can get $Q_h \underline{\otimes} W_r = h + r$.*

As Figure 5(a) shown, we have $res = h + r - t$. And we can see $h + r - t$ and $(h + r) \cdot t$ are similar with each other in function. This also means that TransE ($\|h + r - t\|$) and DualE ($Q_h \underline{\otimes} W_r \bullet Q_t$) are similar when we meet the parameter settings above. That is to say, TransE is a special case of DualE.

Lemma 9 *Double homogeneous projection is equivalent to single projection.*

Proof *As shown in the Figure 5(b), h_1 is the projection point of h on the A_1 hyperplane, and t_1 is the projection point of t on the A_1 hyperplane. Translate the hyperplane A_1 to a position that can cover t. At this time, the projection of h on the hyperplane A_2 is h_2, and the projection t_2 of t is itself. It can be seen that the distance between h_1 and t_1 is the same as the distance between h_2 and t_2. Because the transformation above we made only changed the absolute position of their projection, and did not change the relative position, so this is in line with the principle of embedding our knowledge graphs. So the double different projection can be projected by translating the hyperplane to coincide with t in it, so that it can be converted into a single projection.*

Lemma 10 *Double different projections are equivalent to single projections.*

Proof *As shown in the Figure 5(c), t_1 is the projection of t on the hyperplane A_1, and h_1 is the projection of h on the hyperplane B_1. Let len be the distance vector between h_1 and t_1. Translate the hyperplane A_1 to a position that can cover t, and then the projection t_2 of t is itself. At this time, we reach a new point by shifting t along the vector len by a corresponding length, and we record it as h_2, and this point is the projection of h on the new hyperplane B_2. Taking the vector formed by h_2 and h as the normal vector and passing the point h_2 as a condition, we can calculate the hyperplane A_2. Because the transformation we just made just changed the absolute position of their projection, and did not change the relative position, so this is the principle of embedding our knowledge graphs. So the double different projection can be projected by translating the hyperplane to coincide with t in it, so that it can be converted into a single projection.*

Why are the double homogeneous projection, the double different projection and the single projection equivalent, but the first two are still better than the latter in embedding? This is because there is a premise that it is not easy to find the good projection hyperplane in the latter (the conditions of the equation that meet the conditions are more demanding), but it is really not difficult to find two hyperplanes that are equivalent to it.

Definition 8 *The 4×4 real matrix representation of the quaternion Q has the form*

$$Q = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & -q_1 & q_0 \end{bmatrix}$$

Definition 9 *The 8×8 matrix representation of the dual quaternion Q_d has the form*

$$Q_d = \begin{bmatrix} Q & Q_\epsilon \\ 0 & Q \end{bmatrix}$$

where Q and Q_ϵ are the matrix forms of the quaternion type.

The 8×8 matrix representation of the dual conjugate dual quaternion $\overline{Q_d^*}$ has the form

$$\overline{Q_d^*} = Q^* - \epsilon Q_\epsilon^* = \begin{bmatrix} Q^* & -Q_\epsilon^* \\ 0 & Q^* \end{bmatrix}$$

Lemma 11 *Single projection model is a special case of DualE.*

Proof *The unit dual quaternions Q_d and $\overline{Q_d^*}$ are expressed as definition 4 and definition 5. If we multiply the unit dual quaternions $\widehat{P}_d = Q_d P_d \overline{Q_d^*}$ in 8×8 matrix form, we get the 8×8 matrix*

$$\hat{p}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \hat{p}_1 & \hat{p}_2 & \hat{p}_3 \\ 0 & 1 & 0 & 0 & -\hat{p}_1 & 0 & -\hat{p}_3 & \hat{p}_2 \\ 0 & 0 & 1 & 0 & -\hat{p}_2 & \hat{p}_3 & 0 & -\hat{p}_1 \\ 0 & 0 & 0 & 1 & -\hat{p}_3 & -\hat{p}_2 & \hat{p}_1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

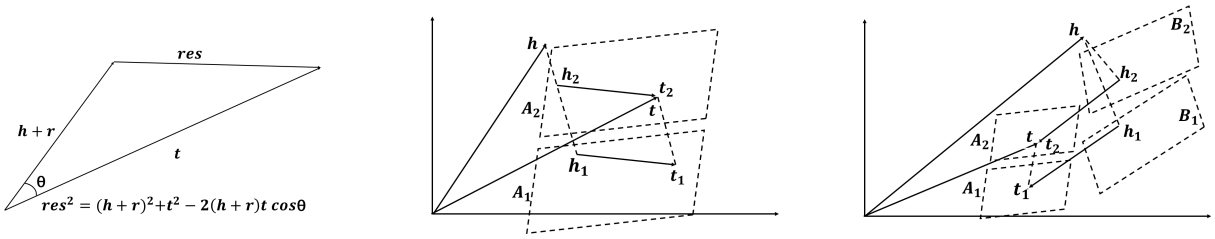
where

$$\begin{aligned} \hat{p}_1 =& q_0^2 + q_1^2 - q_2^2 - q_3^2 + 2[q_1 (p_2 q_2 + p_3 q_3) - q_1 q_{0e} + \\ & q_0 q_{1e} - q_0 (p_3 q_2 - p_2 q_3) + q_3 q_{2x} + q_2 q_{36}] \\ \hat{p}_2 =& p_2 \left(q_0^2 - q_1^2 + q_2^2 - q_3^2\right) + 2[q_1 q_2 + p_3 (q_2 q_3 - q_0 q_1) \\ & - q_2 q_\alpha + q_3 (q_0 + q_{1e}) + q_{2e} - q_1 q_3] \\ \hat{p}_3 =& p_3 \left(q_0^2 - q_1^2 - q_2^2 + q_3^2\right) + 2[p_2 (q_0 q_1 + q_2 q_3) - q_3 q_{1e} \\ & - q_2 (q_0 + q_{2x}) + q_1 (q_3 + q_{2x}) + q_0 q_3] \end{aligned}$$

This unit dual quaternion can be written as $\hat{P}_d = 1 + \epsilon(\hat{p}_1 \boldsymbol{i} + \hat{p}_2 \boldsymbol{j} + \hat{p}_3 \boldsymbol{k})$, i.e., the point $\boldsymbol{p}$ with the position vector $\boldsymbol{p} = (p_1, p_2, p_3)$ is rotated and then translated to the point $\hat{\boldsymbol{p}}$ with the position vector $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$.

The projection matrix can project a vector of a certain space on a hyperplane or space to form a projection point. By assigning parameters to our dual quaternion matrix above, we can achieve the same operation as the projection matrix.

For some other models such as TransH, TransR, TransD, etc., their projection methods belong to double homogeneous



(a) The triangle formed by $h + r$ and t.

(b) The double homogeneous projection is projected into a single projection.

(c) The double different projection is projected into a single projection.

Figure 5: Schematic diagram of single and double projection mapping.

projections or double different projections, so we can directly apply our lemmas to prove that they are special cases of DualE.

Next we take TranA as an example to prove that it is also a special case of DualE.

$$
\begin{aligned}
&(h+r-t)^\top M(h+r-t)\\
=&(h+r-t)^\top PP^\top(h+r-t)\\
=&\left\|P^\top(h+r-t)\right\|_2^2\\
=&\|(p_1^\top(h+r-t),p_2^\top(h+r-t),\cdots,p_m^\top(h+r-t))\|_2^2\\
=&\sum_{i=1}^{m}\left(p_i^\top(h+r-t)\right)^2\\
=&\sum_{i=1}^{m}\|p_i\|_2^2\|h+r-t\|_2^2\cos^2\langle p_i,h+r-t\rangle
\end{aligned}
$$

After the transformation above, we can see that the essence of the TranA model is the inner product of $h + r - t$ and p_i. And it is about the projection of $h + r - t$ on p_i. This operation can also be achieved by the lemmas above.

Proof of Lemma 7

Proof *The dual part contained in the DualE model is not available in other rotation models, and the space where the rotation model is embedded is a subspace of the dual quaternion space. So it can be clearly judged that the rotation family models cannot derive the DualE model.*

Proof of Lemma 8

Proof *The DualE model inherits the good properties of the QuatE model, so it can model the key patterns such as symmetry, anti-symmetry, inversion and composition. However, the translation family models cannot model all the key patterns. Therefore, translation family models cannot derive DualE model.*

Suppose that TransX represents a wide range of TransE's variants in (Parcollet et al. 2018), such as TransH (Wang et al. 2014), TransR (Lin et al. 2015), and STransE (Nguyen et al. 2016), where $g_{r,i}(\cdot)$ denotes a matrix multiplication with respect to relation r. For example, TransE cannot model the symmetry pattern because it would yield $r = 0$ for symmetric relations; TransX can infer and model the symmetry/antisymmetry pattern when $g_{r,1} = g_{r,2}$, e.g. in TransH, but cannot infer inversion and composition as $g_{r,1}$ and $g_{r,2}$ are invertible matrix multiplications.

Experimental details

Number of epoch. The number of epochs need of DualE, QuatE and RotatE are shown in Table 6.

Table 6: Numbers of epochs needed of DualE. QuatE and RotatE.

Datasets	WN18	WN18RR	FB15K	FB15K-237
DualE	2000	20000	5000	5000
QuatE	3000	40000	5000	5000
RotatE	80000	80000	150000	150000

Different variants of Scoring function. Combined with our discussion in the experimental analysis, we study the scoring function of normalization after adding constraints $\langle Q_h \underline{\otimes} W_r^\dagger, Q_t\rangle$ and the scoring function where the relationship is regarded as a weight $\langle W_r, (Q_h \underline{\otimes} Q_t)\rangle$. The results are shown in Table 7.

Link prediction results without type constraints of QuatE and DualE. In order to study the effect of adding type constraints on the experiment, we conducted a study on DualE and QuatE with the same number of parameters, and find that the performance of DualE can also surpass QuatE without constraints. The results are shown in Table 8.

Hyperparameters for DualE without type constraint. We use two versions of DualE in the experiment, which is with/without type constraints. The results are shown in Table 9 and Table 10.

Number of free parameters comparion. We compare the numbers of parameters of TorusE, RotatE and QuatE with that of DualE. And we find that DualE can surpass these models while using fewer parameters. The results are shown in Table 11.

The impact of regularization strategy. We report the MRR of DualE on WN18RR and FB15K237 as follows. We can

Table 7: Analysis on different variants of scoring function. Same hyperparameters setting as before.

Analysis	WN18		FB15K		WN18RR		FB15K-237	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
$\langle Q_h \otimes W_r^{\dagger}, Q_t \rangle$	0.944	0.960	0.766	0.889	0.476	0.577	0.355	0.543
$\langle W_r, (Q_h \underline{\otimes} Q_t) \rangle$	0.563	0.821	0.766	0.890	0.201	0.324	0.319	0.491

Table 8: Link prediction results without type constraints of QuatE and DualE.

Model	WN18				FB15K			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
QuatE	0.949	0.960	0.954	0.941	0.770	0.878	0.821	0.700
DualE	0.951	0.961	0.956	0.945	0.790	0.881	0.829	0.734
Model	WN18RR				FB15K-237			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
QuatE	0.481	0.564	0.500	0.436	0.311	0.495	0.342	0.221
DualE	0.482	0.561	0.500	0.440	0.330	0.518	0.363	0.237

Table 9: Hyperparameters for DualE without type constraint.

Dataset	k	λ_1	λ_2	neg
WN18	200	0.03	0.03	10
FB15K	100	0.03	0	10
WN18RR	100	0.15	0.15	2
FB15K237	100	0.1	0.1	10

Table 10: Hyperparameters for DualE with type constraint.

Dataset	k	λ_1	λ_2	neg
WN18	200	0.03	0.03	10
FB15K	100	0.03	0	10
WN18RR	100	0.25	0.25	2
FB15K237	100	0.1	0.1	10

Table 11: Number of free parameters comparion.

Model	TorusE	RotatE	QuatE	DualE
Space	$\mathbb{T}^k$	$\mathbb{C}^k$	$\mathbb{H}^k$	$\mathbb{H}_d^k$
WN18	409.61M	40.95M	65.53M	65.53M
FB15K	162.96M	31.25M	26.08M	26.08M
WN18RR	-	40.95M	32.76M	32.76M
FB15K237	-	29.32M	11.64M	11.64M

find that the regularization strategy can improve the model.The results are shown in Table 12.

Table 12: The impact of regularization strategy.

Model	FB237	WN18RR
DualE w regularization	0.365	0.492
DualE w/o regularization	0.254	0.403

The impact of loss function. We add a new mean-squared-error loss under the same setting, and the results are as follows. MSE stands for the mean-squared-error loss and CE stands for the cross entropy (CE) loss. We use the MRR metric and we can see it justifies the adoption of CE loss. The results are shown in Table 13.

Table 13: The impact of loss function.

Datasets	MSE	CE
WN18RR	0.430	0.482
FB15K237	0.296	0.330

The impact of different model versions. We consider the ablation where type constraints are not employed but N3 regularization is performed. The different versions of DualE on the FB15K237 dataset are as follows. The results are shown in Table 14.

Table 14: The impact of different model versions.

Model	DualE w/o type	DualE w/o type w N3	DualE w type
MRR	0.330	0.368	0.365